CSMA with Implicit Scheduling through State-keeping: A Distributed MAC Framework for QoS in Broadcast LANs

A Dissertation
Presented to
The Academic Faculty

By

Shantanu Kangude

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

Georgia Institute of Technology

July, 2004

CSMA with Implicit Scheduling through State-keeping: A Distributed MAC Framework for QoS in Broadcast LANs

Approved by:

Dr. John Copeland, Advisor

Dr. Raghupathy Sivakumar

Dr. Henry Owen

Date Approved: May 13, 2004

ACKNOWLEDGEMENT

It gives me immense satisfaction to complete this doctoral thesis. In this space, I would like to thank all who have directly and indirectly helped me achieve this goal, and made the last five years of my life memorable.

I was blessed to have Dr. John Copeland as my thesis advisor. I thank him for all the guidance and help throughout the dissertation research. His encouragement and his faith in us have always pushed us to work harder and aim high. Besides his professional excellence, his patience, humility, and wisdom make him a great mentor and a role model. I am also blessed to have a friend and mentor in Dr. Matthew Sherman. Writing research papers with him has been a very rewarding experience for me. I thank him for his continued guidance, and for all our interactions that have helped me learn a lot about my research area. I would like to thank my dissertation committee members for their valuable time and suggestions. Thank you Dr. Owen, and Dr. Sivakumar, Dr. Lanterman, and Dr. Santos.

The successful completion of this dissertation owes a lot to my colleagues and friends at Georgia Tech. I have learnt a lot from both the casual chats and the serious discussions with them. They help create an excellent environment, here at Georgia Tech. A special thank you to Aravind Nayak who has, on multiple times, reviewed my papers, and given valuable suggestions on difficult problems. I would also like to thank Ms. Kathy Cheek, who takes care of our administrative affairs like a magician. Finally, I would like to thank all my friends and family, who have always supported and encouraged me. Their love and friendship remains the most prized possession of my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CSMA	Carrier Sensing Multiple Access
MAC	Medium Access Control
QoS	Quality of Service
CSMA/ISS	CSMA with Implicit Scheduling through State-keeping
CSMA/CD	CSMA with Collision Detection
CSMA/CA	CSMA with Collision Avoidance
RTS	Request To Send
CTS	Clear To Send
CAR	Collision Avoidance and Resolution
ERG	Explicit Request-Grant
IS	Implicit Scheduling
R-Aloha	Reservation Aloha
CW	Contention Window
DFS	Distributed Fair Sharing
SSA	Slot Signal Assertion
LAN	Local Area Network
MAN	Metropolitan Area Network
OOB	Out of Band
IB	In Band
ENT	Efficient N-ary Tree
CD	Collision Detecting

NCD	Non Collision Detecting
Home PNA, HPNA	Home Phoneline Networking Alliance
PMF	Probability Mass Function
PDF	Probability Density Function
CDF	Cumulative Density Function
TX_OP	Transmission Opportunity
IPSD	Internet Packet Size Distribution
FI	Fairness Index

SUMMARY

With the extensive focus on Quality of Service (QoS) in computer networks in the recent times, there has also been significant interest in designing QoS capable medium access control (MAC) protocols. The two main objectives for QoS that are specific to MAC protocols are channel access fairness and utilization efficiency. While the utilization efficiency requirement is identical to the objective in classical MAC design, the fairness requirement is more comprehensive. Channel access is expected to be rule-based fair (for example weighted fair) across different priority classes of traffic, and equitable across nodes within each class. For bursty and unpredictable traffic in networks, fairness and efficiency involve a mutual tradeoff with the currently popular mechanisms. In this project, we seek to investigate if memory and/or processing may be traded off to achieve fairness along with improved efficiency. We seek to design a QoS MAC framework that achieves this through extensive state-keeping based on the MAC evolution, and intelligent access rules based on the state kept. Carrier Sensing Multiple Access (CSMA) networks, with their *instant* channel feedback, are ideally suited for such a framework. In this project, we propose and evaluate such a framework called the CSMA with Implicit Scheduling through State-keeping (CSMA/ISS) framework.

CSMA/ISS involves the tracking of traffic arrival at *active* nodes, the nodes that need channel access frequently enough to warrant their state-keeping for MAC. It also involves implicit channel access grants to different active nodes according to their estimated or tracked queue backlogs and the fair scheduling requirements. The state-keeping and implicit scheduling allow the mechanism to save channel time (efficiency) that may be required for disseminating the access requirements of various nodes and their

access rights according to fairness requirements. A static, hierarchical, and weighted fair access scheme is designed in CSMA/ISS by allowing repeated *rounds* of access that are weighted fairly according to requirements. Weighted fairness across classes is achieved by invoking channel access for each class in a round as many times as its weight. Within each class, all active nodes are allowed equal access through *in-order* channel access based on a looped (circular) list of active nodes. Fine-grained fairness is achieved as a small fixed amount of data is allowed to be transmitted for every invocation.

Although CSMA/ISS is proposed as a distributed control framework, it may be employed in central control protocols too. The distributed control design is aimed at avoiding some control traffic that is necessitated in central control environments. CSMA/ISS may also be adapted to different types of CSMA networks, both wireless and wired. Since it is in part based on information inference from channel activity, unreliability of channels causes loss in its performance. In such networks, CSMA/ISS may be adapted to keep more state based on explicit feedback, compared to that based on channel activity. While the performance gains may reduce in such cases, in CSMA networks, it may always be employed along with classical QoS mechanisms for better performance than classical mechanisms alone.

The ISS framework was modeled and simulated as a QoS capable MAC protocol for a wired fully connected local network environment. In this document, we present the CSMA/ISS framework, the example implementation, and the results of performance evaluation of the example implementation. Significant performance improvements were observed, and the memory and processing trade-off was found to be low to moderate.

CHAPTER 1

INTRODUCTION

Quality of service (QoS) has been a subject of intense research for computer network engineers in the recent years. With the current popularity of Internet, more and more networked applications and services require better guarantees on delay, bandwidth, and jitter than what the classical best-effort packet delivery service can provide. A number of QoS architectures have been proposed to address the requirement. An important aspect of guaranteeing performance for networked applications is that the entire path from one application end to the other be QoS aware and enabled. This implies that all layers, nodes, and networks involved be QoS capable. Almost all of the proposed QoS architectures have been in the network and higher layers. These architectures use the services provided by the data link and physical layers. QoS requirements from data-link and physical layers can be as simple as a good capacity transmission channel and a low transmission error rate in it. For point-to-point links, these requirements are easily satisfied. In broadcast links however, as the transmission channel is shared between multiple nodes, guarantees of channel availability and capacity are not very straightforward to make. The medium access control (MAC) sub-layer is responsible for arbitrating the use of the shared channel between different nodes in such networks. Traditionally, MAC protocols have emphasized only on maximizing the net use of the shared channel, without any attention to the throughput, delay, and delay variation any *individual* flow may incur. While they have been designed to be statistically fair to participating nodes over large time scales, the fairness is *weak* and does not perform well in the short term. With such MAC protocols, it is difficult to guarantee delay, jitter, and bandwidth. Also, most classical MAC protocols are incapable of providing differentiated service to network traffic of different priorities. Thus, recently there has been significant interest in designing QoS capable MAC protocols.

The design of a QoS capable MAC protocol is greatly influenced by the environment it is aimed at. A MAC environment includes the traffic and the channel characteristics of a network. For bursty and unpredictable traffic in a network, the two main goals for MAC QoS, fairness and efficiency, involve a tradeoff with the currently popular mechanisms. All of the currently popular MAC QoS mechanisms need to spend extra channel capacity (or equivalently time) to achieve fairness. The more comprehensive the expected fairness is, the more is the loss in channel efficiency. In this project, we seek to investigate if other resources such as memory and/or processing may be used to achieve fairness with improved efficiency. Specifically, we seek to investigate if a framework with state-keeping based on MAC evolution, and medium access rules based on the state kept may be designed such that fairness and efficiency are simultaneously improved. Carrier Sensing Multiple Access (CSMA) networks, with their *instant* channel feedback, are ideally suited for such a framework. In this project, we propose and evaluate such a framework, which we refer to as CSMA with Implicit Scheduling through State-keeping (CSMA/ISS).

CSMA/ISS, as the name suggests, involves channel access (scheduling) based on the state kept of MAC evolution. It involves the tracking of traffic arrival at *active* nodes, the nodes that need channel access frequently enough to warrant their state-keeping for MAC. It also involves implicit channel access grants to different active nodes according to their estimated or tracked queue backlogs and the fair scheduling requirements. The state-keeping and implicit scheduling allow the mechanism to save channel time that may be required for disseminating the access requirements of various nodes and their access rights according to fairness requirements. Fairness is implemented in CSMA/ISS by allowing repeated *rounds* of access that are weighted fairly according to requirements. A static, hierarchical, and weighted fair scheme is implemented. Priority classes of traffic share the channel according to their respective weights. This is achieved by invoking channel access for each class in a round as many number of times as its weight. In each invocation, the access aims at transmitting a fixed small amount of data. Within each class, all active nodes share equal access. Such fairness is achieved by allowing *in-order* access to different nodes based on a looped (circular) list of active nodes. Again, fine-grained fairness is achieved as a small fixed amount of data is allowed per access. Thus CSMA/ISS is designed for strong fairness with high channel utilization.

CSMA/ISS is designed as a distributed control framework to avoid control traffic (for efficiency) necessitated in central control MAC environments. It may however be employed in central control environments too. CSMA/ISS begins tracking in a default state, and updates estimates based on information inferred both implicitly from channel activity and explicitly from information in packets. CSMA/ISS may be adapted to different types of CSMA networks, both wireless and wired. Since ISS is in part based on information inference from channel activity, unreliability of channels causes loss in performance. In such networks, CSMA/ISS may be adapted to keep more state based on explicit feedback, compared to that based on channel activity. As the requirement for explicit feedback increases, the performance gains decrease. CSMA/ISS may be employed with classical QoS mechanisms, and a MAC protocol with state-keeping support may always be designed to perform better than one that employs only classical mechanisms. Thus, while performance gains are expected in all types of CSMA networks, the maximum gains may be achieved for wired, fully connected networks (everyone hears everyone else). The ISS framework was modeled and simulated as a QoS capable MAC protocol for a wired fully connected local network environment. In this document, we present the CSMA/ISS framework, the example implementation, and the results of performance evaluation of the example implementation. Significant performance improvements were observed, and the memory and processing trade-off was found to be low to moderate.

The rest of this document is organized as follows. In chapter 2, we define and briefly discuss the objectives of MAC design with QoS. Chapter 3 discusses the influence of channel and traffic environment on MAC design, and the currently popular QoS MAC architectures. Contention resolution is discussed in the context of QoS in chapter 4. In chapter 5, we discuss the research problem this project seeks to address, and the motivation for the proposed solution. Chapter 6 describes the proposed CSMA/ISS framework. The framework was modeled and simulated for the case of wired CSMA networks. Chapter 7 presents the example implementation. Performance evaluation results for the example implementation are presented in chapter 8. We conclude with a summary and the scope for future work in chapter 9.

CHAPTER 2

QOS IN MAC: OBJECTIVES

QoS may be defined as the capability of a network or a network element to make and keep promises (guarantees) about its service quality, as measured through parameters like delay, jitter, throughput, error rate, and availability; and using resources efficiently in doing so. Thus, service differentiation and resource provisioning are important parts of QoS. The components of network QoS are classified in [1] as predictable per-hop behavior and predictable edge-to-edge behavior, along with signaling and policies etc. Shared channel networks may be considered a single hop in networked communication and elements of *predictable per-hop behavior* are thus applicable to QoS in them. A predictable per-hop behavior comprises of packet processing that involves *classification, queuing, and scheduling* (CQS) [1], so as to provide the *promised* service to different flows or classes of traffic. While in point-to-point dedicated links, CQS may need to be done only at the network layer with the links simply forwarding packets between the two end points, in shared channel networks, it is not so. MAC scheduling is performed across multiple nodes that share the channel, and CQS is required of the MAC layer too.

Thus, QoS in a MAC protocol may be defined as its ability to share the common channel among the network nodes efficiently, such that guarantees on delay and throughput can be provided to different flows or classes of traffic across the shared channel network.

2.1 Classical MAC objectives

Classically, the main objective of a MAC protocol has been to share one or more channel/s efficiently among a higher number of nodes [2]. This translates to minimizing the channel time spent in any activity besides valid packet transmissions. High channel utilization implies high throughput and low channel access delay when considered *averaged* over all nodes. However, for same values of mean channel utilization, the delays and throughput that *particular* nodes or traffic classes at nodes register, may vary significantly. This introduces another objective: fairness. Classically, MAC protocols have needed to be fair, when their performance is considered over large scales of time. However, classically there have not been classes of traffic, or priorities. Thus, fairness has implied that all backlogged nodes with equal traffic achieve equal throughput, when considered over long enough durations. Thus, we can list two main objectives of classical MAC design: *fairness and efficiency*.

2.2 QoS MAC objectives

The MAC objectives for QoS may also be listed as fairness and efficiency, when abstracted at a high level. While the objective of efficiency means the same as it meant for classical MAC, fairness for MAC QoS is significantly more comprehensive.

Consider some objectives of a QoS capable MAC protocol. The MAC function needs to support multiple classes of traffic. This may be achieved by classifying the MAC protocol data units (PDUs) by their priority class, and queuing them in different queues. *Eight* priority classes have emerged as a standard [3][4] in this domain, probably influenced by DiffServ codepoint classification [1]. Network-wide traffic in a higher priority class should receive a *defined* better service than the service received by any lower class. It may be strict priority scheduling, in which no lower priority traffic is transmitted until all higher priority traffic has been transmitted; or priorities can also share bandwidth according to some *weights* as in weighted fair queuing [5][6][7]. However, Network-wide traffic in any particular priority class should ideally have equal and fair access to the MAC service. The MAC protocol must be able to provide more *predictable* performance too for individual classes of traffic. Given that the above performance guarantees have to be satisfied with the limited bandwidth of the channel, admission control is required [1]. In local networks, subnet bandwidth managers are proposed for this function [1]. MAC protocols may or may not implement these functions. Some support for admission control in the MAC layer is useful for any external modules that may be employed to perform rigorous admission control. Policing is not applicable at the MAC layer, and shaping though not explicitly intended may happen in the course of scheduling [1]. Thus the *basic* service required from a QoS MAC protocol has two main objectives: one, to provide low and more deterministic access delays, and high and more deterministic throughput to individual traffic classes in the network; and two, to efficiently share the channel among traffic classes, so as to provide them differentiated MAC service according to their priorities. These objectives may be summarized as strong hierarchical fairness along with efficient channel utilization.

2.3 Strong hierarchical fairness for MAC QoS

The discussion in section 2.2 suggests two levels of fairness expected from MAC protocols with QoS. At the top level, fairness is expected across priority classes of traffic. Higher priorities should have preferred access to the medium according to a rule. Ideally, this rule should be such that lower priorities are not completely starved of channel access when higher priorities are overloaded with traffic. A weighted fair access [5] across priorities, with higher priorities having more weight, is ideal in such an environment. The second level of fairness is within each priority class, and across all nodes that have packet backlog in the class. Weighted fairness at this second level should involve equal weights to each node. Hierarchical fairness in this manner implies that fairness performance at one level should not affect the fairness performance at the other level.

Fairness, by definition, is with respect to a given distribution of resources. In the MAC context, the resource is channel capacity, and it is distributed over time. Thus, given a period of time, fairness of capacity distribution may be measured using mean throughput (or equivalently utilization) achieved in the period, as the resource. The longer the durations are, the more is the statistical contribution to fairness. Thus QoS, which is about achieving deterministic behavior, may be considered to improve as fairness is achieved over smaller scales of time. Fairness over smaller scales of time is also called strong fairness. Strong hierarchical fairness is thus an important objective of QoS MAC schemes. Channel utilization efficiency, as discussed above, is the other goal.

2.4 The ideal scheduling benchmark

In order to better understand the challenges in MAC QoS design, consider the best performance that may be achieved by any MAC protocol. In an ideal situation, a MAC protocol should cause valid packet transmissions one after another, with possibly (if there is backlog) only a mandatory inter-frame gap (IFG) between them. These transmissions from various nodes should be such that they satisfy the fairness requirements set for the protocol. This scenario is similar to the case of an output link scheduling in a router. Packets from various queues, at an output link in a router, are transmitted one after another such that a given scheduling algorithm is satisfied [1]. MAC scheduling is similar in the sense of the objective, but all the queues are not in the same node. Thus, a MAC scheduling function does not always have instant feedback on the condition of all queues over which scheduling needs to be performed. Some channel capacity is spent in information exchange between various nodes. This causes a MAC protocol to always perform worse than an equivalent router-link scheduling scheme. Such a router link scheduling scheme is thus a candidate for a benchmark best performing MAC protocol. We refer to this benchmark protocol as the *Ideal Scheduling* MAC protocol.

An ideal scheduling MAC protocol operates like a point-to-point link with queues at various nodes as output queues at the link. It schedules under the unrealistic assumption that the network schedules transmissions one after the other as if all queues in all nodes are part of a single node, and their occupancy is known at all times. An Ideal Scheduling MAC protocol is characterized by the scheduling algorithm it employs. For a chosen scheduling algorithm, the performance achieved is the best possible. Given the requirement of hierarchical fairness as detailed in section 2.3, a corresponding weighted fair queuing algorithm [5] is a natural choice for the scheduling algorithm. Figure 1 shows the Ideal Scheduling MAC scenario for a protocol with 8 priority classes. While Ideal Scheduling is unrealizable, but serves as a benchmark against which all QoS enabled MAC protocols may be compared. The objective of a designer should be to achieve as close an operation to such an ideal MAC, as possible.

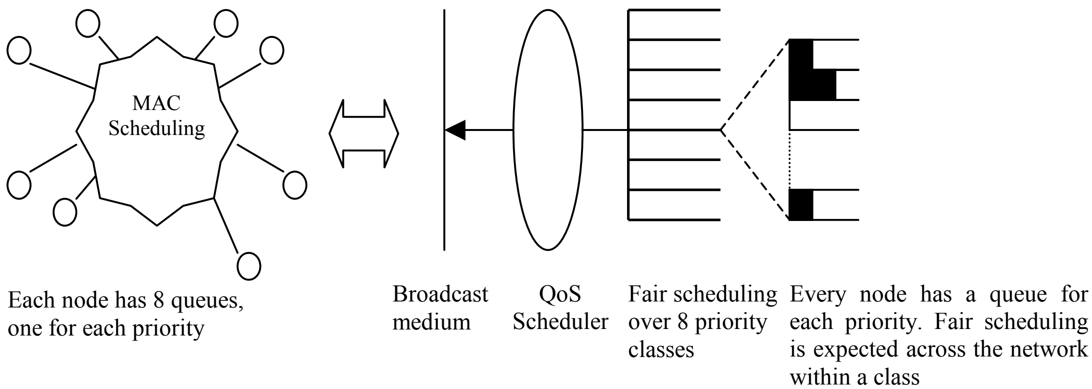


Figure 1. An Ideal Scheduling MAC is equivalent to router output link scheduling

CHAPTER 3

QOS IN MAC: ENVIRONMENT AND ARCHITECTURES

Channels are shared in a network either because of the basic nature of the medium, as in wireless and satellite networks, or because of economic gains in sharing a channel, as in cable networks. In either case, MAC design involves some *given conditions* of channel characteristics and the expected traffic profile (*MAC environment*), and some *choices* about the structure of the MAC protocol. In this chapter, we discuss the affect of MAC environment on protocol design. We also present the currently popular QoS MAC architectures, and their applicability in different environments.

3.1 MAC environments and protocol structure

Before we discuss the influence of MAC environments on the evolution of MAC architectures, consider some of the basic choices that define the structure of a MAC protocol.

3.1.1 Basic MAC design choices

Synchronization structure

Synchronized channel access among different nodes has been proven to be more efficient than unsynchronized access [9]. When nodes interpret the beginning of transmission opportunities (TX_OPs) such that at some location in the network, the transmission delays cause them to be identical, nodes may be considered synchronized. The location at which the instants are synchronized is mostly the intended receiver. Synchronization essentially reduces the number of transmissions that may potentially interfere with each other. For example, slotted Aloha is more efficient than Aloha [9][10].

While synchronization is required in all MAC protocols for efficiency, explicit synchronization effort may not be required in certain networks with propagation delays much lower than packet transmission delays. The beginning and end of transmissions themselves aid in synchronization. This is the case in CSMA networks. Other MAC protocols may need central controllers to be the receivers where the channel is synchronized. Central controllers also aid in the synchronization process itself, through the MAC operation. An example is *ranging* in DOCSIS [11][12]. The choice in synchronization structure lies in the way the MAC protocol allows transmissions. For example, whether variable length transmission segments are allowed or all transmissions are managed in constant size slots. In networks that require explicit synchronization effort, transmission slot sizes have to be predefined for efficiency (and of course for synchronization). In CSMA low propagation delay networks however, transmissions may be variable duration without pre-fixing of durations.

Contention versus reservation

Channel access in a shared channel network is either *contention based or reservation based*. A contention period is a period of time when more than one node may transmit. In a reservation period, the channel is reserved only for a particular node. Between these two extremes may be periods of limited contention and/or reservation. In such periods a small subset of nodes may contend for the channel, or equivalently the channel is reserved for a small number of nodes. Ideally a channel would be best utilized if it were reserved for appropriate nodes all the time. However, there are two main problems. One the reservation of channel itself requires channel capacity. If such reservations are done apriori, they may not reflect the changing channel requirements of different nodes.

Secondly, even if reservations are to be altered over time, the channel requirements or queue information at nodes need to be propagated to the control function. This again requires channel capacity. If frequent changes in channel requirements and reservation results need to be propagated through the transmission channel, its utilization efficiency may be affected. Specially, if a fixed amount of capacity is reserved for each node to propagate such feedback. Contention based access is thus frequently employed for efficiency in bursty traffic environments. Since contention may also cause loss in efficiency due to collisions, mechanisms for reducing and resolving contention are commonly employed.

Centralized versus distributed control

Another MAC choice is that of the location of control in the network. Control may be distributed or centralized. In distributed control (DC) networks, all nodes have the same functionality, and their joint operation determines the transmission rights. A network with a DC MAC can be set up without any extra *infrastructure*, except the participating nodes. Configuration in such networks is much simpler, and ad-hoc networking is possible. There are no centralized points of failure. Also, some control, management, and data traffic, which are necessitated in a centrally controlled MAC, may be avoided in DC MAC. In centralized control (CC) networks, a central node has a different MAC function than other nodes, and it arbitrates channel usage in the network. The main advantage of CC is that complex MAC operation is easily achieved. Processing at non-central nodes is simple. Most state-keeping, processing and data storage are required only at the central node; and explicit reservations are easy to implement. The control structure in a shared channel network may be a natural choice borne out of the network structure, as in cable networks; or it may completely be a matter of choice, as in ad-hoc networks, where a random node may become controller for CC structure, or the MAC may be DC.

3.1.2 Defining characteristics of a MAC environment

Traffic burstiness and unpredictability

Traffic burstiness in a network has a strong effect on the design of MAC protocols for it. The most important factor is the *duty cycle* or the difference in times over which traffic from a node stays constant (or approximately constant) rate and the time required to reserve and un-reserve a channel for a node. Consider for example cellular networks that carry only voice traffic at a constant bit rate (CBR). The time over which the data rate from a node stays constant is much longer compared to the channel time expense in reserving and un-reserving a channel. In such networks the MAC protocol may operate by reserving and un-reserving a channel for transmitting nodes using a (possibly) separate dedicated low-rate channel. A similar MAC structure is efficient even if variable bit rate (VBR) traffic is carried that does not significantly vary in rate and lasts for a long duration compared to channel allocation time. The channel may be reserved for the peak rate in such a case. On the other hand, if significant network traffic is bursty at time scales of the order of the time required for channel allocation, a more contention based MAC protocol may be efficient. Polling architectures for MAC protocols are efficient when all nodes in a polling group have data to transmit[13]. The Internet traffic of today has been shown to be extremely bursty [14][15], and thus networks that carry Internet data have MAC protocols that are more complex than simple 'reserve and access' scheme mentioned above. The reservation methods need to be more efficient and deterministic for QoS in such networks.

Propagation delay versus transmission delay

This is the single most important factor that affects MAC design. Propagation delay may be defined as the time required for a transmission to propagate from one end of the network to the farthest other end. As [16] shows, the network round trip time (RTT) is a time of uncertainty in shared channel networks. In carrier sensing networks [16], the RTT is the minimum period of a slot, the quantum of time that MAC protocols handle, e.g. in backoff. Transmission delay may be defined as the time required to transmit a basic unit of data (packet). If the propagation delay in a network is more than the transmission delay, multiple packets are possible on the medium before all nodes receive the first packet. If the channel is reserved, these multiple transmissions are secure. However, if the channel access is contention based, proper reception of significant amount of data is in doubt. A whole RTT is required for feedback, and depending on the RTT, packet delays may be significant. Even if the propagation delay is less than the transmission delay, but not significantly less, the same problem remains. In general, higher the propagation delay, the more data on the medium that is uncertain, the more time required for feedback, and higher the possibility of low channel utilization.

The ratio of propagation delay to transmission delay, defines a parameter referred to as '*a*' in broadcast networks [16]. As '*a*' increases, allowing only a single packet in the network at any particular time becomes less and less efficient [16]. A cable network [11] is an example of a large *a* network, and the design of MAC protocol is considerably different from that of small *a* networks. Contention based protocols are known to perform much worse than reservation based protocols in large *a* networks [16]. Carrier sensing based multiple access techniques are popular in small *a* networks, as against request/grant techniques in large *a* networks. As discussed in section 3.1.1, explicit synchronization efforts are required for large *a* networks. The synchronization error is the 'uncertain' time, or the time wasted per *unit MAC activity* (a packet transmission or a blank slot etc.). For small *a* networks, this is insignificant, but it is not so otherwise. The channel time in large '*a*' networks is synchronized with respect to *one particular node* referred to as the 'master' node. While different nodes may be masters at different times, changing masters involves channel time expense. Again, in small *a* networks, this expense is much smaller compared to large *a* networks. This suggests a centrally controlled MAC may be efficient for large *a* networks (see *synchronization structure* subsection in 3.1.1). Large *a* networks should be avoided as much as possible by splitting such networks into smaller segments and including 'store and forward' bridges/routers between network segments. However, in certain scenarios, such as in satellite networks, such splitting is not possible.

Channel quality and its variation with time

Some channels vary in their transmission characteristics and capacity with time. A classic example is a wireless channel. With time, as the transceivers or the objects in the environment move, the signal strength at a wireless receiver may fluctuate. Hidden node and exposed node problems are common [17]. These properties are also exhibited in power line networks. In such networks, receiver-based reservations, such as through network allocation vectors (NAVs) set using *request to send (RTS) and clear to send (CTS) messages* in 802.11 networks [18], are efficient. Receiver based reservations notify all potential interfering nodes of channel reservation for a reception and prevent losses.

Due to the highly variable nature of channels, collision detection is not reliable in such networks, and explicit acknowledgements are employed [18]. Concurrency of feedback through channel activity is difficult to achieve, as different nodes may perceive different channel activity due to channel variability. Such channels pose significant difficulties for QoS.

Channel structure, connectivity, carrier sensing, and collision detection issues

Channel structure is another important factor that affects MAC design in shared channel networks. Structure here includes physical and logical connectivity among network nodes, and the transmission and reception channels being the same or different. Transmission and reception may be on the same logical transmission channel or on different channels. If transmitting and receiving channels are different, the network should have a *translator* or a node that converts transmissions from one channel to the other (receiving channel). Thus, such networks are bound to have a central node for translation, and suggest at having a centrally controlled MAC protocol. An example of such a network is a DOCSIS cable network [12]. Depending on the propagation and signal translation delays (in case of different transmission and reception channels), carrier sensing may or may not be employed. Carrier sensing becomes inefficient as delays increase. If the collision detection delay approaches packet durations or more, reservation-based access should be employed as much as possible, and packets with contention-based access should be short.

Physical and logical connectivity in a network significantly affects MAC design. In some networks like satellite networks and IEEE 802.11 infrastructure wireless LANs, nodes are connected to each other through a central node that performs a *bridge* like function (store and forward). All transmissions from non-bridge nodes are directed to the bridge, and receivers are tuned to receive only from the bridge. In other networks like token ring and distributed queue dual bus [16] nodes are connected in a particular manner. Connectivity may suggest the location of control and subsequently influence MAC design. For example, networks that are connected through bridge like nodes may easily employ central control. Directional connectivity may imply certain minimum delays in explicit data exchange between certain nodes. Such delays need to be taken into account for QoS MAC design in these networks. Certain network configurations such as token ring and DQDB rule out contention-based access. QoS in such networks may function by nodes relinquishing their transmission rights according to requirements and MAC rules.

CSMA networks are a class of networks with specific properties [13]. Transmitters and receivers in such networks are tuned to the same channel, and the propagation delays are much smaller than transmission delays. Collision detection may be possible or not depending on the channel quality. Power line [19] and currently popular wireless channels cannot detect collisions and thus collision fragments are complete packet durations of the largest packet. Subsequently collision avoidance is more emphasized in MAC protocols for such networks. In Ethernet and Home Phoneline Networking Alliance (Home PNA) protocols [20] [21], in which collisions are detected and transmission is aborted midway on collision detection, the collision fragments are short in duration. Collision resolution in such networks may be more aggressive (allow more collisions) as collision expense is less. Carrier sensing networks provide the widest choice for MAC protocol design due to their many favorable properties.

Miscellaneous specific conditions

Some shared channel networks have specific characteristics, and subsequently specific objectives from the MAC protocols employed in them. Wireless networks need to be power efficient, and thus expect power control in MAC protocols and QoS mechanisms. A wireless LAN (WLAN) QoS MAC needs to be aware of other co-located WLANs in the same frequency band. It needs to adapt to possible interference from other wireless technologies such as Bluetooth, and random emissions such as from microwave ovens. Newer wireless technologies such as multiple input multiple output (MIMO) systems, directional antenna transceiver systems, and ultra-wideband technologies pose specific challenges, and the MAC solutions are expected to be specifically tailored.

While the actual implementations of access mechanisms in MAC protocols depend on media specific challenges, the utility of basic MAC architectures depends primarily on the above listed MAC environment conditions. Consider next, the basic QoS MAC architectures and their applicability in various MAC environments.

3.2 The basic MAC QoS architectures

Consider the Ideal Scheduling benchmark of section 2.4 again. It is unrealizable in real MAC environments because the control function at any node cannot know the changing states of queues at other nodes in real time, and without extra channel time expense. Thus, two types of feedback are important in MAC protocols: the control function needs to know the arrivals at all nodes, and the transmitting nodes need to know the result of the scheduling function (i.e. which class of which node may transmit). This feedback may be referred to as *request* and *grant* feedback respectively. If all nodes can receive the request feedback from all nodes *reliably*, and within delay bounds (almost concurrently), the grant feedback may not be required to be transmitted. This is so because all nodes may be configured to use a common algorithm to decide the grants, based on identical *request* feedback received at each. Thus, while the grant information may be inferred without communication, the request information generally needs to be communicated. Request and grant feedback may be performed implicitly or explicitly, suggesting the following classification of QoS MAC architectures.

Various mechanisms that are employed for QoS in current MAC protocols may be broadly classified into two basic architectures, as shown in Figure 2: *the Explicit Request-Grant (ERG) architecture, and the Implicit Scheduling (IS) architecture*. The ERG architecture is the most popular architecture for QoS. In it, nodes request bandwidth usage through explicit communication, and explicit permissions may be granted for channel access. This architecture may involve a central controller that receives requests, and grants permissions according to QoS settings. In case explicit grant feedback is employed, a central controller is required, and is designated to perform the function. In the IS architecture, the QoS considerations for channel access are taken into account through the process of the execution of the MAC protocol (activity such as deference, collision, or transmission) itself. It is characterized by absence of any explicit information exchange between nodes, and is popular in distributed control MAC protocols in small *a* parameter networks (low propagation delay compared to transmission delay; see above). MAC protocols may employ mechanisms from both ERG and IS architectures to provide QoS. As we discuss below, the IS architecture is not as powerful or widely employed as the ERG architecture; but is useful for small '*a*' networks.

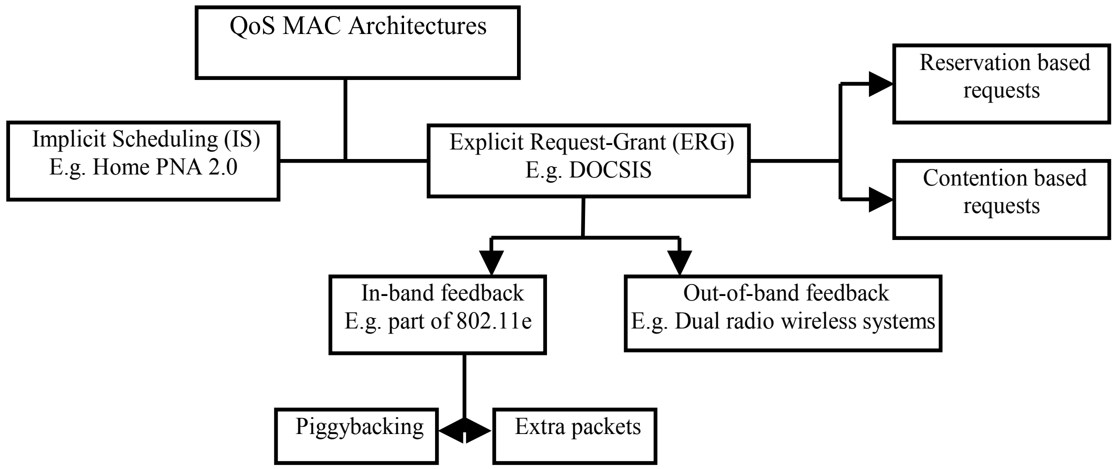


Figure 2. Classification of QoS MAC architectures

3.3 The Explicit Request-Grant (ERG) approach

The ERG approach, as defined above involves explicit exchange of bandwidth requirement, and possibly allocation information. This approach can be used in all types of networks, and can guarantee strong fairness. A drawback however, is the channel capacity used in extra transmissions. Channel utilization efficiency depends on traffic burstiness and the structure of the request/grant feedback. The approach may be considered as an attempt to achieve Ideal Scheduling by sharing information among nodes. A central scheduler may be notified of queue states at all nodes, and it may perform scheduling based on this comprehensive information. Reservation is the prominent mode for data transfer. The approach employs minimum contention, restricted mostly to the transmission of the smaller *request* packets. The ERG approach is thus used extensively in networks with large propagation delays, and networks in which collision detection delays are large. QoS in DOCSIS [12] is an example of the ERG approach.

It may be classified along two different planes, as shown in Figure 2.

3.3.1 Reservation-based versus contention-based feedback

The request-grant feedback in ERG approaches may be reservation or contention based. In a reservation-based channel, each node has some dedicated channel capacity for feedback. Since a reserved capacity for feedback suggests a separate *logical* channel, reservation based feedback may be classified as a form of out of band (OOB) feedback as in 3.3.2. Such an arrangement may waste channel capacity if the traffic is bursty. However, the feedback is contention free and prompt, and the performance in the main data channel is closest to ideal scheduling. In a contention-based feedback channel, nodes contend for feedback in a common channel. While contention improves channel usage efficiency in bursty traffic conditions, the feedback may not be as prompt and fair as in the case of a reserved channel for feedback. In both cases, a fairness-efficiency trade-off is inherent.

3.3.2 Out-of-band (OOB) versus In-band (IB) feedback

If the request-grant feedback in a network is in a channel different from the main data channel, the approach is called out-of-band feedback. OOB feedback requires dedicated separate channel capacity, and may require a costly additional transceiver [17]. The main benefit is that data and feedback transmissions can happen together. Feedback is prompt, and hence the controller function can schedule based on more up to date information. This improves fairness and overall QoS. A dedicated channel for feedback also prevents clustering of request packets that may be caused if they have to wait for long data packet transmissions to be over. Such clustering can cause collisions and wastage of channel capacity. Examples of OOB feedback systems include multi-channel wireless networks [17] and some TDMA based satellite MAC protocols [22].

When the feedback in the ERG approach is in the same channel as the main data transmission, it is referred to as in-band (IB) feedback. IB feedback may be performed either by transmitting extra request-grant packets between data packets, or by piggybacking request/grant information on data packets. Dynamic capacity usage, as in IB feedback may be bandwidth efficient compared to dedicated assigned capacity in OOB. However, the fairness performance is worse as feedback packets need to wait for the completion of data transmissions. Moreover, the capacity saved with the *statistical multiplexing* of data and feedback channel may be more than offset with the loss through extra collisions. Extra collisions may be caused in the feedback transmissions as request packets cluster together while waiting for completion of longer data packets. The main reason though, for the popularity of IB feedback, is the prohibitive cost of employing extra transceivers compared to the benefits achieved.

Piggybacking of request/grant information on data packets is of limited use in burst traffic conditions [23]. Some feedback-packet transmissions may be saved, but exclusive piggybacking is insufficient for QoS scheduling and feedback. Separate packet transmissions or polling may be required along with it [24]. Reference [24] proposes a piggyback feedback with an EOF (end of file) indicator if the packet being transmitted empties the queue.

The DOCSIS cable network [11][12] MAC protocol is an example of ERG architecture using IB feedback. The requests are transmitted to the headend in contention mini slots. Grants follow from the headend to stations, and then data packet transmission follows in data slots. Requests/Grants may be piggybacked too. The proposed 802.11e MAC protocol for QoS in wireless LANs [4] [25] is another example in which a request-grant mechanism is proposed. An interesting IB feedback mechanism, in which the feedback is not deferred till the completion of data packet transmissions, is proposed in [26]. This CSMA/RI mechanism interrupts and resumes ongoing transmissions to reserve channel and extracts the MAC benefits of OOB feedback with IB feedback.

3.3.3 Fairness and efficiency in ERG approaches

Fairness has been studied extensively for QoS scheduling at output links of routers [1]. In the ERG MAC QoS mechanisms, these scheduling algorithms may be used on the queue information generated for the entire network segment through *request* packets. The receipt of request packets in a timely and fair manner from different nodes is the main challenge then. In case the requests are reservation based, or involve contention with extremely low probability of collisions, the request packets from one node do not significantly interfere with or delay the request from other nodes. Comprehensive rule based fairness may be implemented thus. Channel efficiency is affected, as the reserved capacity for feedback packets is never utilized for data transmissions. A fairness-efficiency trade-off is inherent.

In ERG mechanisms that involve *contention* based request packet transmissions that are in-band or otherwise, the collision avoidance and resolution (CAR) mechanism plays an important role in guaranteeing fairness. Since contention based access is part of IS schemes, fairness with contention resolution is discussed in section 3.4.

Channel efficiency in ERG approaches depends on the network traffic profile and protocol structure. Since requests and grants are overhead, the capacity spent for them contributes to inefficiency. If traffic were deterministic over long periods, less frequent request-grants would be required. If contention-based access is employed, efficiency depends on the frequency of collisions and the resolution mechanisms too. Again, traffic profile affects the frequency of collisions, and thus efficiency too. A fairness-efficiency tradeoff is inherent in both ERG and IS mechanisms, as substantiated in the next section.

3.4 The Implicit Scheduling (IS) approach

The implicit scheduling (IS) approach involves all mechanisms that do not employ explicit information exchange. The essence of IS mechanisms are rules that implicitly associate information with current and past channel activity. Information can be inferred through the instances and durations of transmissions and deference, and through state-keeping of MAC evolution. While state-keeping may be equally useful in all types of networks, information inference from channel activity is more useful in small '*a*' parameter networks. This is because, feedback in the form of *type and duration* of channel activity and deference periods, is delayed in large *a* networks. Carrier sensing is also an important form of feedback that is used in IS mechanisms. Thus, IS mechanisms are almost always used for small *a*, CSMA networks. They are also frequently employed in distributed control networks, as ERG mechanisms may require central control for efficient operation, depending on channel type. Contention based access, by definition, belongs to the IS family of approaches. Different contending nodes choose their transmission opportunities individually, and without explicit pre-notification. Controlled contention approaches, in which the number of contenting nodes is limited by explicit notification may be considered to involve both the IS and ERG approaches.

Below are a few mechanisms that may be classified among the IS approaches.

3.4.1 Feedback through instances and duration of channel activity

MAC protocols may operate such that they dictate a certain type of MAC activity at a certain instant of time. They may also dictate such specific activity randomly within certain periods of time, and/or randomly *for* certain durations. Thus, the presence or absence, or the duration of specific MAC activity may imply specific information. The MAC activities involved may be valid packet transmissions, collisions, inactive channel (deference), or signals. While packet transmissions, collisions, and blank slots are well known and understood, signals have been employed in protocols only in recent times. Signals are small packets with periodic pattern of bits, intentionally without any data. They are only employed so that carrier-sensing nodes may detect signal power in the channel and interpret the channel as *asserted*. Collisions of signals do not change the 'asserted' interpretation of channels, and involve no data loss. Next, let us consider some IS mechanisms that involve instances and duration of MAC activities.

Varied deference

Varied deference is a powerful and extensively used mechanism in carrier sensing (CSMA) MAC protocols. Deference is the process of waiting before beginning packet transmission. Since a small period of channel inactivity between transmissions is essential for proper reception in CSMA networks, a minimum deference is mandatory after the end of any transmission. However, significant amount of feedback information may be exchanged through deference beyond the mandatory *inter-frame gap (IFG)*. Such deference may be probabilistic or deterministic. P-persistent CSMA and binary exponential back-off [9] are two of the earliest uses of probabilistic deference. Probabilistic varied deference is inherent to contention avoidance and resolution, as discussed in chapter 4. Fairness may be implemented with varied deference through a bias in the probabilistic deference. If all nodes defer randomly in a similar manner, they statistically achieve equal access. However, if certain nodes probabilistically defer for less time than others, they achieve more access than others. An extreme case of such bias is that of deterministic varied deference.

Deterministic varied deference is used in CSMA based MAC protocols to realize priorities in transmissions. In IEEE 802.11 MAC protocol, the different deference periods for inter-frame spacing cause acknowledgements to take priorities over newer packets, and the Point Coordination Function (PCF) to take priority over the Distributed Coordination function (DCF) (refer to SIFS, DIFS, and PIFS in [18]). In Home PNA MAC protocol [21], lower priority packets wait for longer inter-frame gaps (IFGs) than higher priority packets. Such a scheme implements a strict priority scheduling, as shown in Figure 3. While the absence of any channel activity within a short deference period may imply the absence of any traffic in a particular class, it may also imply other information based on a set rule. The idea may be to test the existence or non-existence of a property across a set of nodes. It may also imply that a reservation is relinquished. This, for example is the case in Reservation-Aloha (R-Aloha) protocol [22].

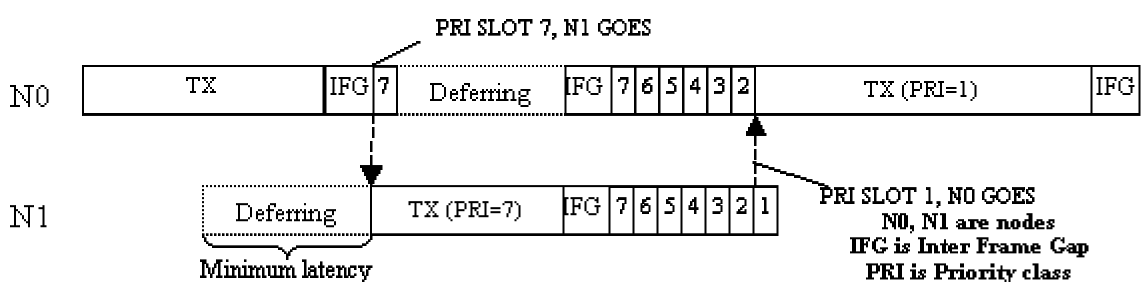


Figure 3. Varied deference priority mechanism in Home PNA MAC protocol (from [3])

Consider next, the general case of biased random deference. MAC time may be interpreted as a sequence of transmission opportunities. A fair access mechanism may be designed such that for every packet, nodes transmit by randomly choosing one transmission opportunity (TX_OP) out of disjoint sets of TX_OPs. A flow may access the shared channel more by choosing smaller sets of TX_OPs. This is the basis of *Distributed Fair Sharing (DFS)*, proposed by Vaidya et al [27]. The DFS idea has also been adopted for IEEE 802.11 E standard for QoS in wireless LANs [25].

The enhanced DCF (EDCF) mechanism in 802.11 E MAC protocol proposes to employ *biased* probabilistic deference to realize a probabilistic priority access scheme. The collision avoidance function of the protocol causes each node to select a random number of slots to defer before (and after) transmission. The random number is chosen from the *contention-window* (CW) number of slots. The lower the CW, the smaller is the deference before (and after) packet transmissions. The 802.11E protocol proposes higher priority packets to have lower CWs and thus have probabilistically prioritized channel access.

Slot signal assertion (SSA)

In SSA, signals are transmitted in specific slots for feedback. The feedback happens through the detection of channel activity. The presence of activity implies one or more nodes have a certain property, while the absence implies none of them possess it. The idea is to cause all nodes with the concerned property, to assert a signal. If multiple nodes perform SSA, channel activity is still detected. However, the number or identities of such nodes are not known. This method is useful for detecting the absence or presence of network wide characteristics; e.g. if all nodes with non-empty queues are caused to assert a signal during a particular slot, the absence of any signal in the slot implies an empty network, and opposite otherwise. As in the case of varied deference, SSA may also imply other complicated form of feedback, based on a pre-set rule. In Home PNA MAC, SSA is employed to signal the queuing of one or more nodes in the tree based contention resolution mechanism [3].

Biased random signal durations (Blackbursts)

A modification of SSA scheme involves transmitting short variable-duration bursts of carrier, called *Blackbursts*, as a means of feedback. This process may be considered the opposite of varied deference, since signal transmissions are variable duration instead of deference periods. Fairness in access is achieved in the same manner as in varied deference. For example, nodes asserting the longest duration signal may *capture* access to the channel. Such a scheme was first proposed by Sobrinho et al in [28].

Transmissions, blank slots, or collisions

In IS mechanisms, regular channel activities such as packet transmissions, collisions and empty slots are made to convey extra information to the MAC function through some set rules. A transmission may imply a reservation for the transmitting node in subsequent frames at the transmitted slot position. This, for example, is the rule in R-Aloha [22]. Similarly, while a collision implies that two or more nodes in a set of nodes attempted transmission, it may additionally imply that certain later slot positions are reserved for the colliding nodes (possibly part of the definition of a collision resolution mechanism). MAC rules may indicate specific subsets of nodes to transmit in specific subsets of TX_OPs. In such a case, blanks slots, valid transmissions, and collisions may imply the absence or presence of packet backlog in specific nodes or subsets of nodes. MAC protocols may be designed so that transmissions and collisions imply complicated feedback in innovative ways.

3.4.2 Information inference through state-keeping

State-keeping may be defined as storing relevant information from the history of the MAC operation in order to make decisions in the present or the future. The temporal evolution of MAC operation may be compactly stored as *state* for making future medium access decisions. Earliest MAC protocols such as Aloha and slotted Aloha [9] [10] keep very little state. Most CSMA MAC protocols however, keep state at least in order to avoid or resolve collisions. For MAC protocols with priorities, additional separate state information may be kept for each priority, and for coordinating transmissions among priorities (e.g. Home PNA [3]). State-keeping may be employed for traffic estimation and prediction, which in turn may aid in QoS aware (fair and/or efficient) access [29] [30]. In the cable network domain, [23] and [30] have proposed the use of recent traffic history for making MAC decisions, specifically to reduce contention. While state-keeping is increasingly being employed in MAC protocols, no comprehensive QoS architectures have yet been proposed. This document proposes one such architecture.

3.4.3 Fairness and efficiency in IS approaches

As in the case of ERG approaches, the currently popular IS approaches also involve a fairness-efficiency trade-off. The two main IS mechanisms of varied deference and varied black bursts, both require longer deference or burst periods for improved fairness. Fine (i.e. over small scales, or strong) fairness in varied deference approaches like DFS require longer contention windows. In these distributed approaches, nodes choose deference or burst periods independent of the choice of other nodes. Thus, in the case of deterministic deference/burst periods, the number of different periods that are chosen by nodes approaches the number of different flows in the network. This requires increasing amount of channel time in deference or Blackbursts, both of which cause unutilized channel time and loss in efficiency. In case biased random deference in employed, the same argument can be extended. As the number of flows over which fairness is expected grows, the number of TX_OPs from which a random TX_OP be chosen to transmit grows too. Another issue with these IS mechanisms is that hierarchical fairness is not straightforward to achieve. A new flow that starts at some node will affect all flows and not just the flows in the same priority class. In order to achieve two levels of fairness, these mechanisms may possibly be modified with more structured deference/bursts. They may also employ some explicit feedback. In either case, hierarchical fairness requires additional unutilized channel time, and thus causes loss in efficiency.

CHAPTER 4

CONTENTION RESOLUTION AND QOS

In order to utilize a shared channel efficiently in bursty traffic conditions, and to reduce latencies during lightly loaded network conditions, contention based channel access is frequently employed in MAC protocols. Contention based access too may not be efficient if there are frequent collisions and collision fragments are long. MAC protocols thus frequently employ minimum contention possible to reserve the rest of the channel time for appropriate transmissions. While the extent of contention employed versus reservation in a MAC protocol may vary, the QoS objectives of fairness and efficiency carry over to the process of contention resolution. In ERG approaches, the request packets may be transmitted through contention-based access. The collision avoidance and resolution (CAR) mechanisms involved should provide prioritized access to request packets of higher priority classes, and coarsely a first-come-first-served (since packet sizes may be different, a later arriving shorter packet may need to be served earlier; see fair queuing in [1]) access to those belonging to the same class. In IS approaches, reservations happen through implicit feedback from MAC operation such as through collisions, deference, and state keeping. Again, CAR mechanisms are important for fairness and access efficiency. In this section, we discuss the different types of CAR mechanisms available, and their properties with respect to the above two QoS objectives. The optimal operation of CAR mechanisms and their applicability to different MAC environments is also discussed in the context.

CAR mechanisms may be considered to operate upon *contending sets* and *collision sets.* A contending set is a group of nodes or flows that *may* collide while transmitting at a particular transmission opportunity. Collisions happen when multiple nodes in such a set have packets to transmit. As the set population increases, the collision probability increases. The main aim of CAR mechanisms is to reduce the set to a group of nodes such that only one of them is set to transmit. A *collision set* may be defined as a group of nodes that *have* collided or *will* collide at a transmission opportunity. Division of contending sets into smaller subsets can be done *statically or dynamically*. Dynamic division forms the CAR mechanisms in MAC protocols. Static division is a permanent division that is useful if the divided sets would collide often otherwise, or if direct interaction between the traffic from different sets is to be avoided. We group the study of avoidance and resolution mechanisms together, as avoidance is a resolution process that begins before a collision.

4.1 Classification of CAR mechanisms

4.1.1 Tree based versus free-for-all

All CAR mechanisms can be broadly classified into two main types [31]. One is the ‘free for all’ ALOHA [9] [10] like approach, in which nodes attempt to transmit (or retransmit) messages hoping for no interference from other nodes (Figure 4). P-persistence mechanisms, in which nodes attempt transmissions during slots (or transmission opportunities) with some probability *p,* belong to this category [31]. The other approach is to split the contending nodes into a tree structure of smaller subsets, with the subsets transmitting in order, one after the other [9] [31] [32]. A tree splitting mechanism divides a collision set into a number of subsets, with each subset of nodes transmitting one after the other in order. The transmission of each subset may be considered as another CAR effort, and recursion is inherent. Tree splitting CR mechanisms have been shown to have better delay, throughput, and stability properties over ALOHA like approaches [32], but require some state-keeping by all nodes. They are popular in QoS capable MAC protocols as they also cause lower variation in resolution times [31].

4.1.2 Open versus closed resolution

CAR mechanisms may also be classified as *closed or open* (Figure 4). A closed mechanism does not allow new nodes to enter a CAR cycle, which is the period in which all the nodes involved in CAR process transmit their packets. Thus, new packet arrivals at nodes during the CAR process are not served until the CAR process is over. A MAC process may be viewed as a series of closed CAR cycles in such protocols. (Some cycles may involve single packet transmissions without any collisions). A first come first served (FCFS) order is maintained with respect to *groups* of packet arrivals in such protocols. Such fairness improves QoS by limiting access delays. Strictly speaking, a tree mechanism is a closed mechanism. However, adaptations of tree mechanisms that are open are frequently employed as we discuss below. Open CAR mechanisms are the opposite, and allow new packet arrivals to enter the CAR process. The random nature of CAR mechanisms may cause new arrivals to get channel access before all the old arrivals are served. An extreme case is the widely deployed Ethernet protocol. Its 1-persistent nature causes a last come first served (LCFS) order of transmissions [16] when colliding nodes are backing off. Closed CAR mechanisms with their fairness are better suited for QoS than the open ones. However, as observed below, in distributed control CSMA environments, they require state-keeping and constant monitoring of channel activity.

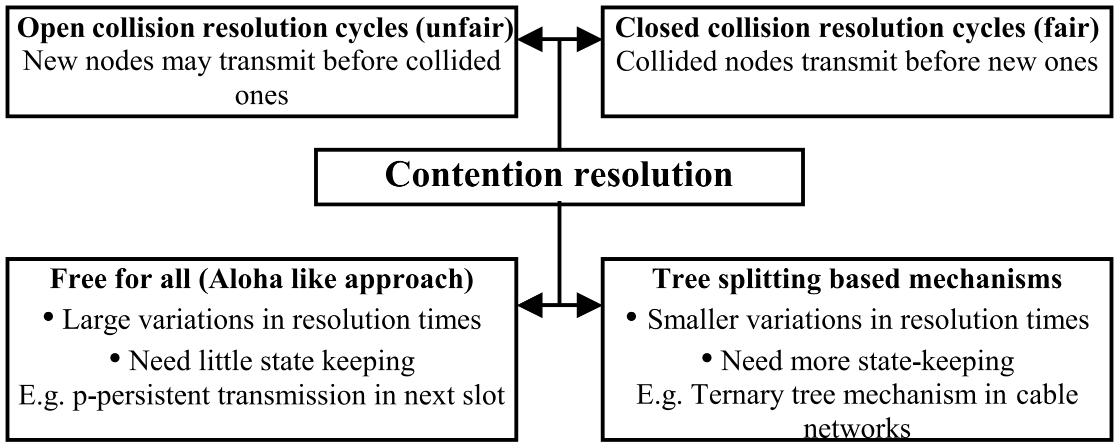


Figure 4. Classification of collision avoidance and resolution mechanisms

4.2 Optimal operation of CAR mechanisms

There has been more than two decades of research in MAC protocols and collision resolution mechanisms. Most of the results documented have been for specific MAC environments with Poisson arrivals and specific adaptations of various CAR mechanisms. References [2], [9], and [16] document these results in a comprehensive manner. Most of the literature discusses Poisson arrivals, and *average* throughput and delays for MAC protocols as a whole. While Poisson arrivals have been shown to be inappropriate to model network traffic [14] [15], MAC protocols have also become more complex in the way CAR mechanisms are incorporated in them. Thus performance analysis results that are specific to CAR operation rather than the MAC protocol as a whole, and based on a given number of contending nodes rather than an arrival rate are becoming more relevant.

Frequently, a CAR stage involves a number of nodes each of which choose one transmission opportunity randomly out of several available. Each transmission opportunity may result in a valid transmission, a collision, or a blank slot. The optimal number of transmission opportunities for a given number of contending nodes depends on the relative durations of blank slots and collision fragments. Packet duration may not be relevant for the evaluation of a CAR mechanism, if the objective is to minimize the mean channel time wasted *per valid transmission*. Other variables that influence such a metric are the number of contending nodes, and the number of transmission opportunities to choose from. In case delays or utilization are involved, packet durations do influence the performance metrics. However, the capability of a CAR mechanism does not depend on packet durations. Before we consider optimality in detail, a brief introduction to the concept of rounds is in order.

The concept of rounds in CAR mechanisms

All CAR mechanisms inherently involve *rounds* of operation. The operation of CAR mechanisms is frequently interrupted and/or interspersed with other MAC activity such as polling and reservation based access, or access to other priorities, or simply inactive periods. While the CAR operation may be *frozen* at and continued from any instant (obviously except midway between slots, transmissions or collision fragments) in most CAR mechanisms, CAR mechanisms inherently too, operate in stages. The evolution of the next stage in these mechanisms is determined based on the result of activity in the current stage. These stages are refereed to as rounds. For example, CAR mechanisms may be required to know if a CAR cycle has ended at a stage, or further rounds are required. In tree mechanisms, nodes need to know if there were collisions that require further tree splitting, or if the CAR cycle is over with transmissions by all nodes in the collision set. Depending on the parameter *a* in the network, such feedback may involve delays as shown in Figure 5. The MAC operation may also further delay the beginning of the next round. These inter round delays involve time expense that is not part of a CAR cycle. That is, given zero feedback delay, rounds may operate without interruption. Thus, optimizing the CAR process for a MAC environment involves two parameters: the total duration over which a CAR cycle is spread, and the total duration of the CAR cycle itself. Optimizing both variables is identical only if there are no delays between rounds. Since inter round times are frequently random, and depend on traffic and other factors, the variables of interest are *the number of rounds*, and the CAR cycle duration. Below we discuss the state of the art in optimality results with respect to these two variables.

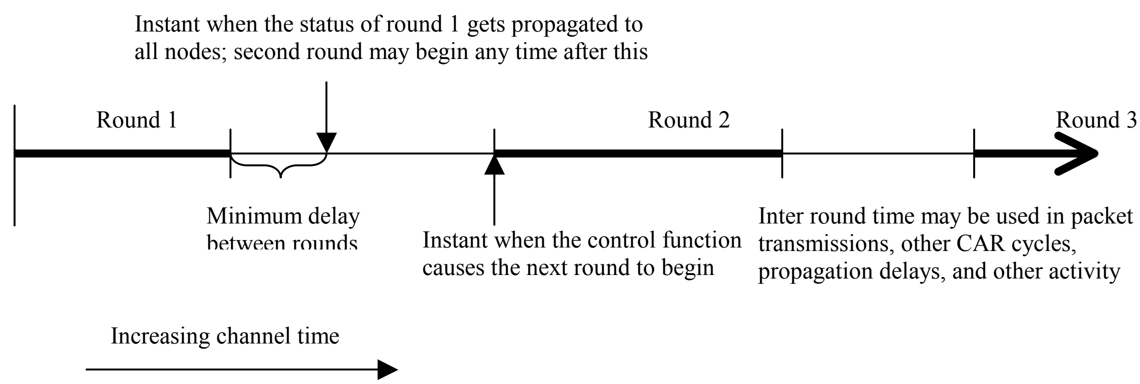


Figure 5. Collision resolution mechanisms inherently operate in rounds

4.2.1 QoS CAR mechanisms with slotted identical duration MAC activity

As discussed in 3.1.1, in large '*a*' networks such as cable and satellite networks, the channel is frequently synchronized for efficiency with a basic medium access unit of a slot (or mini-slot). In such networks, collisions, valid packets, and blank slots are all the size of the slot duration in contention-based access. This may hold true in some small '*a*' networks too, depending on the MAC design, as requests may be transmitted in short packets of the size of a slot. The analysis of CAR mechanisms in such systems is significantly tractable, and is presented below.

Consider the optimality results for minimizing time spent in *a single* stage of CAR without any regard to the overall CAR cycle time. For both Aloha-like and tree-based CAR mechanisms, the operation of a stage of resolution is similar. The results for the overall CAR cycle will follow from the results here. This scenario is similar to Slotted-Aloha [9] with *n* slots and *k* packets (or nodes). It has been shown that in such an environment the slot throughput is maximized for *'n=k'* [23] [33], and the maximum *mean* throughput is

$$T = (1 - 1/n)^{(n-1)} \tag{1}$$

This maximum throughput tends to the maximum slotted Aloha throughput of 1/e as the number of slots (which is equal to the number of nodes) tends to infinity. The corresponding result for p-persistence CAR mechanisms states that given a transmission opportunity, the value of probability *p* that results in the maximum likelihood of a single collision-less transmission is equal to *1/k*, if *k* is the number of participating nodes [9] [16]. It is evident here that a reliable estimate of the number of contending nodes is useful in operating a CAR mechanism optimally.

Given the '*n=k*' optimality for one stage of resolution, the optimality over the entire CAR cycle depends on the inter-round evolution of the CAR mechanisms. In Aloha-like mechanisms, there is no splitting and all the unresolved packets form a single set which undertakes another round of resolution process. If the number in the new set is estimated reliably, each round of resolution may employ the same '*n=k*' optimality result. Frequently though, estimation of the number of nodes in a collision set is difficult. The estimate, if employed, may be updated even within a round. For example, based on the MAC feedback within a round in the form of further collisions, transmissions, or blank slots. One method of estimation is to assume a larger set if more collisions occur. This implies either the number of slots chosen is increased or equivalently the probability of transmission during an opportunity is decreased. Binary exponential backoff (BEB), which was developed to stabilize Slotted-Aloha retransmissions [9] [34], is frequently employed in CAR mechanisms (Aloha-like ones). BEB is discussed more in the next subsection. If the estimation of collision multiplicity is reliable to an extent, tree based protocols are better employed, given their favorable properties. Equation (1) also suggests that CAR performance is better for smaller collision sets (smaller n in (1)). Thus mechanisms that cause resolution on smaller collision sets most of the time are desirable. Tree mechanisms that subdivide sets into smaller sets fall into this category.

While '$n=k$' optimality discussed above yields smaller durations of CAR cycles in the mean sense, it does not necessarily translate to minimizing the number of rounds or the *variation* in the CAR cycle duration. In some cases, minimizing the CAR cycle duration is not the ultimate goal. Some increase in the mean CAR cycle duration can be tolerated, if the number of rounds and the variation in resolution times can be significantly reduced. The trade-off is generally case specific. Slot size is usually small and increasing the number of slots per stage improves the number of transmissions resolved per stage. This is true for both aloha-like and tree based mechanisms. For p-persistence approach, the equivalent is to reduce the probability p. The premium on reducing the number of rounds depends on the inter-round times. If feedback is delayed as in large 'a' networks, the larger and the more random the delay between rounds, the more important it is to reduce the number of rounds. The number of rounds required is a monotonically decreasing function of the number of slots employed per round. Thus, the more the delay between rounds, the better it is to employ more slots in a round. Sometimes, the inter-round time is used for transmission of data packets (in both small '*a*' [35] and large '*a*' networks [12]). The number of slots per round then should be such that enough *request* packets are successful in a round for data packet transmissions to occupy the channel in the inter-round time. Another important objective is to reduce the variance of CAR cycle periods. Less randomness in resolution times implies more fairness and better QoS with tighter guarantees. Some optimality analyses for fixed degree tree splitting mechanisms for minimizing CAR cycle variance are presented in [36] and [37]. The degree of tree splitting (DTS) (or equivalently, the number of slots per round) that minimizes variance is observed in general to be larger than the DTS that minimizes mean. Thus, increasing the number of slots beyond the optimal number for minimum mean periods is beneficial in terms of reducing both, the number of rounds and the variance. We observe that there may be three distinct objectives: minimizing mean CAR cycle duration, minimizing the variance of CAR cycle duration, and reducing the number of rounds of resolution. Depending on the traffic type expected, the performance objectives, and the type and latency of feedback, a reasonable compromise may be chosen.

Optimality results for tree-splitting mechanisms

Consider some optimality results for tree-splitting mechanisms. The splitting rule may be *static* and pre-set, or *dynamic* based on the feedback after each round. Some mechanisms that use feedback *within* a round to steer the rest of the round are discussed for small a distributed control networks in [9]. These are however not in popular use because of their tendencies to be unstable, and the requirement that the channel be highly reliable.

Consider dynamic splitting first. If the degree of splitting can be changed at every stage, possibly based on the feedback from the previous stage, the 'n=k' optimality result may be used for each round for the entire cycle. A reliable estimation of the multiplicity of each collision subset is necessary for a useful dynamic operation in this case. A maximum likelihood estimation and slot allocation (equivalently splitting) scheme, based on the number of colliding and successful slots, is proposed in [23] for cable networks. Centrally controlled MAC is ideal for such a dynamic tree splitting mechanism as the controller can arbitrate the splitting at each stage for each subset. In distributed control environments, if the physical layer is reliable enough so that with a high probability the MAC evolves identically for all nodes involved, a common rule for dynamic splitting may be used by all of them. However, dynamic splitting is not employed if estimation of multiplicities and dynamic changing of DTS are difficult or unreliable. Let us consider such cases next.

Static or preset splitting may take one of the following two forms. If the mechanism is operated in a closed manner such that the beginning of a new CAR cycle is known throughout the network segment, the DTS for each round may be pre-decided. Such a method is more general than the second method in which the DTS is set to a constant number for entire CAR cycles. For example, ternary splitting (DTS=3) is employed in Home PNA [3]. For both the forms of static tree splitting, the optimal tree results for minimum mean CAR cycle period are available. Consider the single round optimal splitting result. At the first stage, the DTS is optimal at the same value as the number of contending nodes (the *'n=k'* optimality). After this stage, some nodes may transmit successfully, while others may collide. Each collision forms a new subset for resolution. It may be shown that each such collision subset is highly likely to have two nodes. Intuitively, if the number of slots is the same as the number of contending nodes, every collision implies at least one blank slot. The odds are against having collisions with high multiplicities and a large number of blank slots. If the subsets are highly likely to have two nodes, any further collisions are even more likely to have only two nodes involved. Thus, a DTS of 2 is ideal for subsequent rounds of splitting. The above intuitive argument is in agreement with formal analysis presented in [32]. Reference [32] proves that a tree that causes minimum mean CAR cycle period is one that is split with a degree almost equal to the initial collision set in the first round, and with a degree of two in the subsequent rounds. The slot throughput in this scheme may be improved if instant feedback is available within a round [9] [32]. As mentioned earlier, some of them suffer from instabilities and require extremely reliable channels. Tree mechanism may also be made *open* to give *unblocked stack algorithms* [9] [32].

With the failure of Poisson traffic arrival assumption for today's networks, initial collision multiplicities in collisions may be difficult to estimate. In distributed algorithms, changing the DTS frequently is difficult. In such cases, the MAC follows a constant DTS throughout a cycle. Mean value analysis for such a mechanism with packets, slots and collision fragments of identical duration may be derived from the analysis in [35]. The main aim of the algorithm in [35] is to maximize the net data throughput while very short request packets were transmitted in contention based mini-slots interspersed with data packets. The analysis assumed the request packets and the slot size to be significantly smaller than data packets and emphasized on reducing the number of rounds. The larger the number of slots used per splitting, the lower is the number of rounds. Splitting as low as ternary was shown to achieve significantly high (of the order of 1) utilization for Poisson arriving traffic. A more general analysis presented in [37] allows for slots, collision fragments and packets to be of any duration. The analysis also considers optimal splitting with respect to minimizing the variance in CAR cycle periods. As we discuss in the next subsection, optimal splitting depends on the relative sizes of collision fragments compared to empty slots. For slots and collision fragments of identical constant duration, low DTS from 2 to 5, depending on the collision multiplicities, are optimal for minimizing mean CAR cycle periods. For minimizing variance, the optimal DTS are somewhat higher [37]. The optimal DTS do not keep increasing with increasing collision multiplicities because of the quick subdivision into smaller sets in tree mechanisms.

MAC protocols are ideally designed to involve as little contention as is efficient, and collisions of high multiplicities are expected to be rare. If this is not the case, it is best to redesign the MAC protocol. Given the likelihood of low multiplicities in collisions, ternary splitting has been popular in many MAC protocols with fixed DTS [3] [35] [38]. While binary splitting may be optimal for high slot throughputs (if multiplicity is 2), ternary splitting reduces the number of rounds required. However, the Home PNA 2.0 MAC protocol with ternary splitting has been shown to operate sub-optimally [21]. A careful analysis should be done before a choice of a constant DTS is made in tree mechanisms. QoS objectives of low variability in resolution times and a small number of resolution rounds should also be accommodated in the decision process.

In certain MAC environments, closed tree mechanisms may not be the best to implement in spite of their favorable properties over Aloha like mechanisms. A constant channel monitoring is required in tree mechanisms to track the evolution of CAR cycles. Closed mechanisms and other mechanisms that involve state-keeping based on channel activity may not function best if the channel is unreliable and/or varies frequently. In such environments, open tree mechanisms (unblocked stack algorithms [9]) and hybrid of Aloha-like and tree mechanisms are frequently employed. An example is the IEEE 802.11 protocol as discussed in the next sub-section. The design of such hybrid mechanisms may employ contention resolution and fairness results discussed above.

4.2.2 QoS CAR mechanisms with non-identical duration MAC activity

In small '*a*' CSMA networks, the most efficient CAR arrangement involves blank slots, collision fragments, and packets of different durations. Since the propagation delay is small, it takes a short interval to detect the absence of any transmissions during a transmission opportunity. Even the smallest packet (possibly a *request* packet) may be larger than this short duration. Thus blank slots are significantly smaller in duration than valid packets or collision fragments. A collision may or may not be detectable by the transmitter, depending on the medium characteristics. If the transmitter is able to detect collisions and abort its transmission thereafter, the resulting collision fragments are mostly of constant (with a small variable jitter to account for random detection delays) short duration. These fragments may be somewhat larger than, but are of the same order of magnitude as the blank slots. In case the transmitter is unable to detect collisions, packet transmissions continue, and the channel is occupied till the largest packet in the collision completes its transmission. Assuming a short feedback time (small '*a*'), a negative acknowledgement or an absence of acknowledgement establishes the collision on the medium for all transmitters. Thus, the collision fragment duration is significantly larger than blank slots, and about the size of the longest packet duration in the collision. We refer to the above two scenarios as collision detecting (CD), and non collision detecting (NCD) cases. Consider next, the optimality results for both. The analysis is simpler for the CD case, as collision fragments are independent of packet sizes.

A useful metric for the evaluation of efficiency of a CAR mechanism is the unutilized channel time (all time except valid packet transmission time) per valid packet transmitted in CAR cycles of given collision multiplicities. We refer to it as *normalized collision expense (NCE)*. Revisiting the QoS objectives, a CAR mechanism is expected to: minimize the mean and variance of NCE, and reduce the number of rounds required for resolution. Reducing the net CAR cycle period is identical to reducing the number of rounds in small 'a' networks, as the feedback is prompt and the inter-round time is not constrained by the channel to be long. However, the inter-round time may be dictated by the number of CAR engines that are simultaneously operating (possibly for different traffic classes), and their time interspersed operation. In case it is required to reduce the number of rounds in the CAR process, the method remains the same: increase the number of transmission opportunities (or the DTS) per round. The rest of this section concentrates on optimality results for minimizing the mean and the variation of NCE.

The extensive research on CAR mechanisms for the case when slots, collisions, and valid packet transmissions are of identical and constant duration may be extended for the CD and NCD cases of this subsection. Consider a single stage operation. The 'n=k' mean-optimality with n slots and k nodes is valid for collision fragments and blank slots of identical constant durations. For the case of general relationship between blank slot size and collision fragment size, a portion of the analysis presented in [21] and [37] may be employed. Given *n* and *k*, the analysis leads to the numerical computation of the mean net temporal spread of a single round of resolution. A useful result would be one that provides optimal *n*'s for minimizing mean and variance of NCE in one round of resolution, given *k* and the relation between blank slot size and collision fragment size. However, such a result is not documented to the best of authors' knowledge. The analysis results of [21] and [37] are for tree splitting mechanisms that employ a fixed DTS throughout a CAR cycle. Intuitively however, we may extend the '*n*=*k*' result. Since it is unrealistic to expect collision fragments to be smaller than blank slots, consider increasing sizes of collision fragments versus the blank slot size. In such a case, trading off a collision to a blank slot improves channel utilization. This may be achieved by increasing the number of slots (or DTS) per round, thereby reducing the probability of collisions. Thus, as collision fragment size increases as compared to slot size, the optimal *n* for a given *k* also increases beyond the '*n*=*k*' result. If such an *n* is known for any given *k*, and *k* is reliably estimated at every round of resolution, a dynamic tree mechanism that employs the optimal *n* in every round may be implemented. An optimal pre-set static tree mechanism may similarly be designed. No analysis is available though, that provides the optimal pre-set trees for general relations between slot size and collision fragment size, to the best of authors' knowledge.

For tree mechanisms with fixed DTS throughout a CAR cycle, an analysis of mean and variance of NCE is provided in [37]. A similar analysis for the Home PNA CAR mechanism is provided in [36]. For different adaptations of the tree mechanism, different values for variables like inter-frame-gap, slot size, collision fragment size, and packet durations may be substituted in the relations derived in [37], and the corresponding behavior can be characterized. For each such adaptation, the analysis provides results to optimize the DTS with respect to other variables in the CAR process. The general trends for both CD and NCD networks were observed to be the same. The existence of optimal DTS's with respect to minimizing the mean and variance of NCE was established. They were not observed to be identical, suggesting that a compromise based on relative importance to mean and variation may be required. The loss in CAR performance as DTS was increased beyond the optimal value was observed to be significantly less than that if DTS was decreased below the optimal value. This suggests that if there is a possibility of estimation error in collision multiplicity, the expected performance loss is less if the chosen DTS is higher rather than lower [37]. Ternary splitting, as employed in Home PNA 2.0 MAC protocol has been shown to be optimal only for a limited number of cases [36]. Considering the NCD and CD cases, the NCD case is always less efficient, as expected. The optimal DTS for the NCD case are always higher and vary more with the collision multiplicity than those for the CD case. Increasing packet sizes cause more CR expense in the NCD case and require higher DTS for optimal operation. The reader is referred to [37] for complete results. Dynamic CAR mechanisms, that employ fixed DTS throughout a cycle but may change the DTS for different CAR cycles, may employ the analysis results of [37], and are proposed therein.

As collision fragments increase in size, the higher optimal value of n for minimizing mean NCE per round is useful in minimizing randomness in resolution too. With respect to maximizing the throughput in terms of the number of utilized transmission opportunities per offered opportunity, the '$n=k$' optimality holds, and leads to high likelihood of a collision multiplicity of 2 in case further collision happen (as discussed in the previous subsection). As n increases, the probability of collisions in a round decreases, and any collisions that happen are even more likely to have a multiplicity of 2. Thus further resolution is even more deterministic. The mean number of successful transmissions in one round also increases, and the number of rounds required for resolution decreases. Overall, the values of n that minimize the mean NCE may also be useful in causing low variations in resolution times.

Open tree mechanisms or hybrid of tree and Aloha-like mechanisms are frequently employed in small 'a' distributed control networks in case the transmission channel is unreliable, or constant state-keeping and channel monitoring are to be avoided. Binary exponential backoff in Ethernet [20] may be considered as having a binary splitting property. Ethernet backoff does not *freeze* when a transmission is detected midway. So, in the case of collision multiplicity greater than 2, the first valid transmission can be long enough to exhaust the back-off timers of all other stations. These stations will definitely collide as they attempt transmission together (1-persistence is commonly used). Thus BEB in Ethernet is not a *recursive* binary tree splitting mechanism. The tree depth is always one. The IEEE 802.11 MAC [18] improves over Ethernet by freezing the backoff on channel activity detection and continuing it once the activity is over. Nodes may be considered to randomly queue themselves up in a distributed manner in one place out of a *contention window* (CW) number of random opportunities. BEB is adopted so that on collisions CW is doubled. There is no recursive splitting, and hence the protocol may be considered as an open tree mechanism with a depth of 1. The tree like property of the mechanism to defer before and after a transmission, and the suspension of the backoff on transmission detection, lends it fairness [4]. Reference [27] lists various methods to employ the CW approach to fairness. As discussed in chapter 3, the fundamental idea is that CW range may have different lower and upper limits to constrain the frequencies of access achieved by various flows. While the open nature of such open/hybrid mechanisms makes them less fair compared to closed tree mechanisms, such mechanisms are a good compromise given the dynamic and unreliable nature of channels such as the wireless and powerline channels.

CHAPTER 5

THE MOTIVATION FOR CSMA/ISS

In the previous chapters, we discussed the current state of the art in the design of QoS mechanisms in MAC protocols. While various mechanisms have been proposed and employed for MAC QoS, there is an inherent tradeoff between fairness and efficiency in each. In this chapter, we discuss the motivation for state-keeping as a means of improving fairness along with efficiency.

5.1 Fairness Efficiency Tradeoff in classical approaches

The fundamental cause of the fairness efficiency tradeoff lies in the unpredictable nature of network traffic. Loss in channel utilization efficiency occurs when channel time is spent in feedback through packets, deference, collisions, or signals. Frequent feedback is required if traffic characteristics change frequently, leading to loss in efficiency. Fairness, by definition implies allowing opportunities for everyone. If such opportunities are not used, loss in efficiency occurs. This, for example, is the source of inefficiency in IS mechanisms such as varied deference and blackbursts. The longer contention windows and burst durations are required for fairness so that a larger number of flows are accommodated. As finer fairness is required, the longer durations of deference and bursts also allow for finer resolution in a fair transmission order. If these durations were small, the fairness bias in the random choice of transmission opportunities would not be realized. Collisions are also more likely. Thus, in almost all classical distributed approaches to fairness, finer fairness implies lower channel utilization. In the classical ERG approaches, the channel capacity used for the transmission of request and grant packets is wasted as far as data transmissions are concerned. For fine grained fairness, the control function in an ERG approach should have an accurate knowledge of traffic arriving at different nodes in different flows. If the traffic profile changes frequently, more control packets are required and efficiency suffers. Thus, we observe that in both ERG and IS classical approaches, there is a fairness-efficiency tradeoff.

5.2 Traffic estimation and implicit grants: the scope for improvement

Given the classical ERG and IS mechanisms, and the fairness-efficiency trade-off, a natural question is *where* is the scope for improvement. We observed that the inefficiencies in fairness implementations occur because of the unpredictable and bursty nature of traffic. If traffic were predictable over long periods of time, requests and grants for channel access would be required only infrequently. The access grants can be issued according to fairness rules, and thus fairness would be implemented efficiently. The main source of inefficiency is the frequently changing nature of channel requirements of various nodes. Contention based request or data packets are used frequently to avoid the inefficiency of blind reservations for all nodes. However, depending on the traffic, contention based access may also be inefficient with a high incidence of collisions. The main problem is for the control function in MAC protocols to know the varying channel requirements of various flows.

Estimation of channel requirements of flows by tracking their recent history of transmissions can be a promising approach to achieving fairness with efficiency. While estimation of requirements can substitute for ‘request’ packets and save capacity, implicit access grants can substitute for ‘grant’ packets. An implicit access grant is a channel access right interpreted based on a common access rule and identical state information at all nodes. An implicit access grant does not use any extra capacity, and the common access rule can be a fair scheduling algorithm. While such implicit access grants and traffic estimation may save on channel capacity usage, they cannot be expected to be as accurate as ERG approaches. Thus, any scheme that employs them would need *correction mechanisms* to ensure high performance. These correction mechanisms may involve some explicit and/or implicit feedback. Regardless of the feedback type, if implicit grants are employed, the access mechanism is contention based. This is so because implicit grants at different nodes may not be identical in case of estimation differences. Implicit grants, however, may not be employed at all. For example, estimation may be employed along with polling (explicit grants). While estimation and tracking alone may improve performance, further improvements may be achieved with implicit grants by avoiding 'grant' type control traffic.

5.3 Distributed access in CSMA networks: The ideal environment

As discussed in 5.2, traffic estimation and implicit grants methods have significant potential for QoS performance improvements. Since implicit grants scheme is inherently contention based, and contention is by definition distributed in nature, the maximum gains from the scheme are expected in a distributed contention based access environment. While a distributed access method is employable in most MAC environments, some MAC environments are better suited for gains from estimation and implicit grants.

The ideal environment for improving performance with estimation and implicit grants is one in which the estimation mechanisms work well, and the correction mechanisms (see 5.2) can be most efficiently implemented. Any correction mechanism or an estimation mechanism based on channel access history would perform better if the feedback delays were small compared to the transmission delays. Thus, small *a* parameter networks are the ideal candidates. While explicit feedback involves a fixed amount of channel capacity usage in all type of networks, implicit feedback is most efficient and rich in CSMA networks. Various IS approaches to QoS are based on the potential of rich implicit feedback in CSMA networks as discussed in chapter 3. Thus, small *a* CSMA networks are the best candidates for exploring performance improvements with traffic estimation and implicit grants. The *CSMA with implicit scheduling through state-keeping (CSMA/ISS)* MAC QoS framework is thus designed for distributed contention based access in CSMA local networks. The basic concept and the method of traffic estimation and tracking proposed in CSMA/ISS may however be employed for centrally controlled and ERG environments too.

5.4 The ideal scheduling emulation idea for CSMA/ISS

The design of CSMA/ISS is inspired by the benchmark ideal scheduling MAC protocol described in chapter 2. Ideal scheduling is based on the perfect knowledge of all queues at all nodes, at all times, as in the case of output link scheduling in routers. CSMA/ISS is based on keeping state *of* all active nodes *for* all active traffic classes *at* all nodes. The idea is to generate a scenario similar to the one at output links of routers. If all nodes maintain identical such queue states and employ a common scheduling algorithm, the scenario is similar to ideal scheduling. The main challenge is to maintain and evolve identical states at all nodes, and maintain fairness and efficiency even in case of errors.

The CSMA/ISS design is based on generating a dynamic state of the network at all nodes, and scheduling based on such state. Given that such a state is generated and updated throughout MAC operation, CSMA/ISS aims to achieve strong hierarchical fairness by round robin access to different nodes and classes based on their weights. For each priority class, it seeks to dynamically evolve a MAC state that tracks the queues and the arrival rates at active nodes in that class. An independent medium access state is maintained for each traffic class. Fairness across priorities may be achieved by invoking the MAC operation for individual classes as many times as their weight, and by serving equally at each invocation. Continuous such operation can achieve weighted fairness across priorities. In order to achieve fairness within a class, each active node may be served equally in a round robin manner. The rest of this chapter motivates the design of CSMA/ISS in further detail.

5.5 Implicit information inference for fairness with efficiency

The estimation and tracking of network wide queues, and implicit grants in CSMA/ISS involve information inference in the normal course of MAC activity. That is, information inference without extra channel capacity usage, or with negligible capacity usage. This may be achieved by information inference through normal *MAC activity such as instants and durations of packets, slots, collisions, and signals; and also through a negligible overhead explicit feedback piggybacked* in normal packets. In order to infer extensive information, possibly intelligent state-keeping may be employed. Thus, possibly memory and processing may be traded off for simultaneous improvement in fairness and efficiency.

Consider some methods through which implicit information inference is achieved for MAC QoS in CSMA/ISS.

5.5.1 Network traffic patterns and tracking of active nodes

The estimation and tracking of queues in CSMA/ISS approximates a slowly varying constant arrival rate scenario for queue filling. This approximation is based on the characteristics of network traffic observed and documented comprehensively in recent times. Network traffic may be classified as being of three types.

1. **Bulk data transfer**

   Bulk data transfer type traffic involves a large enough bulk of data that requires some time for complete transmission over the network. The data amount is fixed and ready at the transmitter, and does not change in real time as transmission proceeds. The time over which this bulk data is transferred over the network is significant enough to consider the transfer a 'flow' that affects MAC evolution. Affecting MAC evolution implies that the *history* of the *flow* may be useful for MAC scheduling. Bulk transfers are characterized by *on* periods in which the transfer rate is close to a constant depending on the minimum capacity link from one end to the other. Majority of today's network traffic can be classified as bulk data transfer type. It includes the World Wide Web traffic, and all download and upload type traffic such as ftp, and multimedia downloads and uploads. Traffic characteristics of web and all download/upload type traffic have been studied and documented comprehensively in [14], [15], [39], and [40]. These references have shown that such traffic can be modeled as having on-off periods. In off periods there is no traffic, while in on periods the traffic arrival rate may be approximated as constant rate. The durations of on-off periods were found to be heavy tailed, and the traffic itself self-similar [39] [40].

2. **Extended interactive/streaming traffic**

A traffic type somewhat different from bulk transfer type traffic is the interactive or streaming traffic that varies in real time as transfer is in progress. Examples include interactive voice and video traffic, and *live broadcast* type of traffic. Such traffic is variable rate, and the duration of a *flow* is significantly larger than the time scale over which the rate changes. In this type of traffic too, tracking transmission history can be useful for intelligent medium access.

3. **Scattered short data bursts**

Besides the above two traffic types, networks also carry traffic that is sporadic in nature. Traffic that involves short bursts of the order of one or two packets, every once in a while. The number of packets in a burst is so small, and the inter-burst time so large that no MAC performance improvements can be achieved by tracking of arrivals in such networks. The flows end before any significant tracking can be done. Examples include telnet and email traffic. These resemble bulk data transfer, but the bulk amount is small and sporadic, and is not of significance for MAC performance.

On-Off periods and slowly varying constant rate approximation

Given the above types of network traffic, we observe that all flows may be approximated as having *'on-off'* periods, with a slowly varying constant traffic arrival rate in on periods. While scattered short data bursts do not fall in this category, their influence on MAC performance is insignificant.

Tracking active nodes and their queues

The slowly varying constant rate approximation for flows in *on* periods, suggests at the possibility of tracking the arrival rate and queue sizes at nodes. Since the knowledge of queue occupancy of active nodes has significant bearing on the performance of a MAC scheduling function, tracking active nodes and their queues is an interesting proposition for MAC QoS. We define an active node in a traffic class as a node that is in an *on* period with respect to that traffic class. Next, we explore how the tracking of queues may be performed, and how detailed information may be inferred implicitly.

5.5.2 Ordered access, estimation updates, and state evolution

Consider a scenario in which all nodes in a CSMA broadcast network segment have identical state information and a common access rule. Let us assume the following.

1. The identical state information is in the form of an *ordered* list of nodes with their queues and arrival rates tracked.
2. The access rule is such that a contiguous subset of nodes in the above ordered list are allowed to randomly chose one of a contiguous set of transmission opportunities (TX_OPs) to transmit. See Figure 6.
3. Contiguous subsets of nodes in the ordered list are mapped to contiguous sets of TX_OPs for transmission, as in Figure 6.
4. Each subset of nodes that is mapped to a set of TX_OPs as above has only a *single* node that is *estimated* to be backlogged (the gray circles in Figure 6).
5. Each set of TX_OPs that the subsets of nodes are mapped to, has a fixed number of TX_OPs. For example, the number is 3 for the case in Figure 6. A fixed number of TX_OPs are employed because the number of estimated backlogged nodes in the subset of nodes is also fixed (equal to 1; see number 4 above).

6. Every packet transmitted is broadcast with an explicit 1-bit feedback that declares whether the tracked queue of the transmitter lags or leads (or is equal to) the real queue of the transmitter.
7. Based on the single bit feedback, all nodes update their estimates for the transmitting node. They increase or decrease the filling rate of the estimated queue using an identical algorithm.

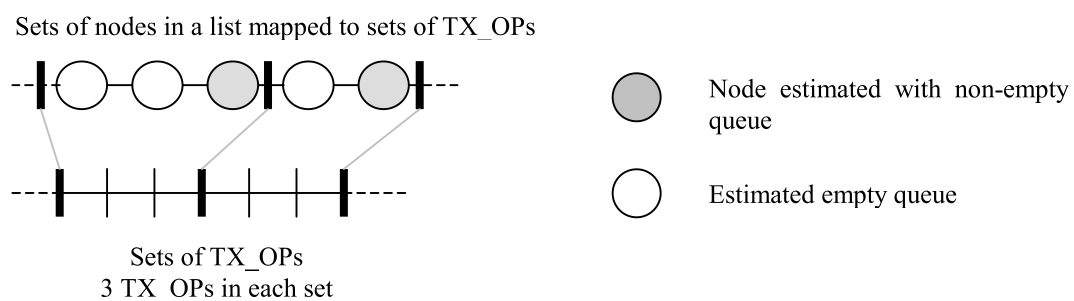


Figure 6. Mapping of ordered subsets of nodes to ordered sets of TX_OPs in CSMA/ISS

Given the above operation, there is significant information that can be implicitly inferred. If the tracked state matches the real queues at nodes, then in every set of TX_OPs, only a single TX_OP would be used, and used for valid packet transmissions without any collisions. However, since the estimated/tracked state of nodes' queues cannot be expected to match the real state perfectly, the MAC activity and transmissions in TX_OPs provide valuable feedback to correct the estimates. Consider the following occurring in the TX_OPs.

1. Valid transmission by the node estimated to be backlogged in the subset: It substantiates that the tracking is not altogether different from reality. The one bit explicit feedback in the header tells whether the estimated queue size is smaller or

bigger than the real queue size. The estimated queue-filling rate can be increased or decreased based on the feedback.

2. Valid transmission by a node other than the one estimated to be backlogged: The queue estimate of the transmitting node is wrong, so corrections can be made. The estimated queue-filling rate is increased.
3. No transmission by the node estimated to be backlogged: The estimated queue state of the given node is incorrect. So corrections are made to the state. The estimated queue-filling rate is decreased.
4. Collision of packets: Multiple nodes in the subset of nodes are backlogged. Additional transmission opportunities are introduced for collision resolution. A closed collision resolution cycle occurs before the next subset of nodes in the ordered list is allowed to transmit. On valid transmissions of collided nodes, the identities of the nodes are known. Based on the explicit 1-bit feedback in the packet headers and the identity of the transmitters, estimates are updated based on the points 1 and 2 of this list.

The above described method of state-keeping and channel access may be implemented to dynamically track the queues at various nodes, and allow fair, in-order access. The method described above may be independently implemented for the 8 priority traffic classes in the network. The rest of this chapter describes how hierarchical fairness and efficiency are achieved in CSMA/ISS.

5.5.3 Fine grained fairness within a traffic class

Fairness within a traffic class may be achieved by round robin channel access to all active nodes. CSMA/ISS aims to achieve such fairness by ordered access to all active nodes as described in 5.5.2. The ordered list of nodes mentioned in 5.5.2 is close looped in the form of a ring. This ring is traversed continuously in one direction to allow channel access to backlogged active nodes in a round robin manner. Thus the MAC evolution of every class incorporates *rounds* of equal access to active nodes. The granularity of fairness is proposed to be the maximum transmission unit (MTU) [13] for the network. Thus, every time a node accesses the channel, it may send up to MTU amount of data. Since the sizes of packets to be transmitted may not add up to closely match MTU, in order to not lose fairness, the borderline packet needs a special rule. Such a packet, whose transmission would cause the net data serviced to increase beyond MTU, is transmitted or not probabilistically. The probability of transmission is inversely proportional to the excess data beyond MTU that would be transmitted in one access if the packet were transmitted. The inverse probability causes about equal amount of data to be transmitted per access on *average*. The inverse probability idea is similar to the idea of having longer contention windows for longer packets in the Distributed Fair Sharing (DFS) scheme by Vaidya et al [27]. The ordered access while traversing the ringed list of nodes ensures fine fairness. The order is maintained as small contiguous subsets of nodes that are mapped to contiguous sets of TX_OPs. The collision resolution mechanism is proposed to be *closed* for fairness. Further fairness and efficiency in collision resolution is achieved by inducing order in the transmissions during collision resolution. This is explained next.

5.5.4 Efficiency through collision avoidance and fast resolution

Since the estimation process does not use up any extra capacity, and since CSMA/ISS attempts to emulate ideal scheduling based on the estimates, the performance achieved by ISS depends on the accuracy of the estimates. CSMA/ISS attempts to allow

TX_OPs for estimated backlogged nodes. Thus, it saves channel time spent otherwise in blank slots and collisions. It avoids TX_OPs wasted on nodes that are not backlogged. Since every subset of nodes that is allowed to contend includes only one node that is estimated backlogged, collisions are effectively avoided if the estimates are good. Moreover, the contention sets are expected to be statistically similar regardless of the number of nodes that might be backlogged in the network. In classical methods, the MAC becomes less efficient as the number of backlogged nodes increases. We observe that state-keeping helps in reducing contention for efficiency.

While CSMA/ISS reduces contention as described above, it also speeds up collision resolution substantially. It is the inherent order from the list (ring) of nodes that lends determinism to collision resolution. As described in chapter 4, every *stage* (round) in most collision resolution mechanisms involves nodes choosing one TX_OP randomly from a number of TX_OPs. Multiple such stages are required whenever multiple nodes choose the same TX_OP resulting in collisions. In the random choice of TX_OPs, there are chances of collisions. However, if the choice is deterministic such that different nodes choose different TX_OPs, collisions can be resolved in the first stage itself. As Figure 7 shows, CSMA/ISS provides such deterministic mapping. A subset of nodes that is mapped to a set of TX_OPs may be considered ordered from one end to the other in the ring. If there is a collision, each colliding node may choose a TX_OP in collision resolution corresponding to its number in the set. Thus, each node can choose a distinct TX_OP in the first iteration itself. While such ordered access saves channel capacity wasted in further collisions, it also lends the protocol more fairness in collision resolution. The order of nodes listed in the ring is the order in which the nodes should transmit for fair access. This is the order in which the nodes transmit even in collision resolution, as outlined above. Thus, we observe that collision resolution in CSMA/ISS is not only efficient, it also aids in the protocol being fair. In spite of a contention-based access method, CSMA/ISS is designed to avoid collisions, and to efficiently resolve collision that might happen. The access method is designed to be fair too.

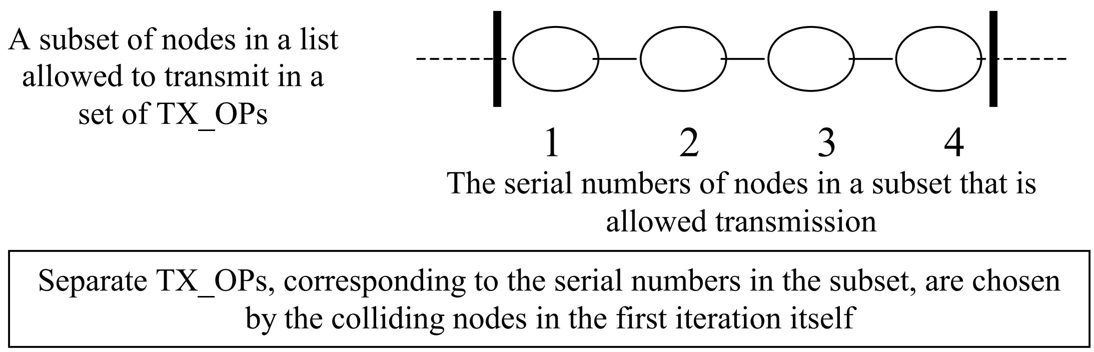


Figure 7. The order in the list of nodes in CSMA/ISS aids in collision resolution in the first iteration

5.5.5 Fine grained fairness across priority classes

The multi-priority operation of CSMA/ISS is motivated by the standard processor sharing approach of serve and pause. The approach involves independent MAC evolution for each priority class, and invocation and stalling of the MAC operation for each class with time. Such an approach has been employed in MAC protocols earlier too. For example in the Home PNA 2.0 MAC, eight classes have independent MAC state in the network [21]. In the approach, only one class's MAC is serviced at a time. State variables are used to store the state of every class's MAC evolution. These variables allow a serve and pause type of operation for each class. Fine-grained fairness across priority classes is achieved in CSMA/ISS by employing such a serve and pause access method.

CSMA/ISS aims at *static and fixed* weighted fairness across 8 priority classes. A set of weights is chosen for the 8 classes with higher priorities having more weight than lower priorities based on any chosen rule. Just as in the case of fairness within classes, fairness across classes is also implemented by having repeated rounds of access. In each round, every class's MAC operation is served according to their weights. This is achieved by serving and pausing each class's MAC operation, a number of times equal to its weight. Each invocation of a MAC operation is served until a fixed amount of data has been served or there is no backlog. This fixed amount of data is suggested to the MTU for the network, as discussed in the single priority operation of CSMA/ISS. For finer fairness, the invocations for each class should be interspersed uniformly. Overall, multi-priority operation may be visualized as attempting round robin access to different classes with every round having possibly multiple accesses for each class, depending on their weights. Chapter 6 explains this in more detail.

In this chapter, we described the motivation behind the design of CSMA/ISS. We described how the ideas of tracking queues of active nodes and ordered fair access based on the estimated states may be employed for QoS in CSMA networks. Next, we formally describe the CSMA/ISS framework.

CHAPTER 6

THE CSMA/ISS QOS FRAMEWORK

In this chapter we describe CSMA/ISS framework in detail. It is important to note that CSMA/ISS is a framework that can be employed to design a MAC protocol. The framework does not define protocol implementation details, which may be chosen depending on transceiver and channel characteristics. Implementation details include inter-frame gap (IFP) period and collision resolution mechanism's operating parameters besides other parameters. The framework however, does provide guidelines on making all design decisions.

6.1 The target environment

As discussed in chapter 5, the target environment for CSMA/ISS is *distributed contention-based access in small 'a' parameter CSMA networks*. While the design described is the most efficient *distributed* control design, the estimation and tracking mechanisms may be employed in centrally controlled environments too. The only core requirement for CSMA/ISS is a small 'a' parameter CSMA network. CSMA/ISS can be employed as a part of a QoS MAC scheme, or an entire MAC protocol in itself. It may be adapted to various CSMA environments, both wired and wireless. Since it involves feedback that is both implicit through access activity and explicitly piggybacked with packets, the amount of implicit versus piggybacked feedback can be chosen such that the protocol suits the channel characteristics. As the reliability of full connectivity reduces for a channel (as with hidden nodes [41]), the reliability of implicit feedback reduces, and thus more of the piggybacked feedback is required. Some framework parameters may also be chosen to adapt CSMA/ISS to specific environments. Section 6.5 describes in some detail, the manner in which CSMA/ISS may be adapted to different channel characteristics. We will observe that the performance improvements with CSMA/ISS are expected to be the maximum in the case of wired fully connected networks.

6.2 A single priority operation

In this section, we describe the independent state-keeping and MAC operation that is prescribed for each priority class in CSMA/ISS.

6.2.1 State-keeping per traffic class

As mentioned in chapter 5, CSMA/ISS aims to track the queues and arrival rates at all active nodes. The state for each active node is stored in a data structure called a *state record*. A state record stores the following fields of information.

1. Address: This field stores the MAC address of the node whose state is stored in the record.
2. Estmtd_q_size: This field stores the estimated queue size for the corresponding node. The size is a non-negative integer value denoting the number of packets estimated in the queue of the node being tracked.
3. Tick_period: This is the time interval after which the Estmtd_q_size field is incremented by 1 to emulate the arrival of a packet in the queue tracking process. Thus, Tick_period is the inverse of the rate at which the queue is filling. The field is an integer value that denotes the number of clock ticks between two increments in the Estmtd_q_size field. This number is copied to another variable and decremented every clock cycle till it reaches 0, which causes the increment of Estmtd_q_size by 1.

4. Prv_indx: This is a pointer to the next state record in the ring (looped linked list) of state records of active nodes. This field is employed for ring management.
5. Nxt_indx: This is a pointer similar to Prv_indx above, and stores the address of the next record in the ring. This field is again employed for ring management. The concept of *next* and *previous* gives a sense of direction of traversal, in the state ring.

The *state ring*, as shown in Figure 8 is a closed looped linked list of state records that is employed for state-keeping of active nodes in the network. When set, the ring always involves 2 or more state records, up to a maximum value. The maximum number of records maintained may be referred to as MAX_STT_KEPT for the ring. It is an important variable as it denotes the amount of possible state-keeping possible for the MAC protocol. For example, if MAX_STT_KEPT is just 2, then the protocol is essentially devoid of any state-keeping gains. An important property of the state ring is that it always has a state record called a *Dummy* record. The dummy record does not involve any state-keeping, but is employed to allow nodes that are not in state to transmit so that they do not interfere with the nodes that are already in state. As we discuss in the next subsection, CSMA/ISS allows different nodes to access channel based on the records traversed in the state ring. As the dummy record is traversed, nodes that are not in the ring can transmit.

Other state variables employed in CSMA/ISS include a *currently scheduled record (CSR)* and a *previously scheduled record (PSR)* (Figure 8). These are records, the nodes of which are estimated to be backlogged, and form the border of a subset allowed to transmit. The subset allowed to transmit begins from the next record of PSR and ends at the first estimated backlogged node while traversing the ring. This first estimated backlogged node is the CSR, as it is the node expected to transmit from the subset. PSR is excluded from the subset, but it is the CSR of the previous subset allowed to transmit.

The operation of a collision resolution mechanism requires its own state variables such as back-off counters. These are also part of the state kept per priority. Miscellaneous bookkeeping variables, specific to implementations, may also be required in CSMA/ISS.

6.2.2 State ring traversal

The access mechanism in CSMA/ISS operates by allowing access to specific subsets of nodes from the state ring, one after the other in continuous TX_OPs. As mentioned in chapter 5, a fixed number of TX_OPs are allowed per subset. The number of TX_OPs allowed per subset is a design parameter. If the estimation process is expected to be reliable, 1 TX_OP per subset should be enough. However, if the collision fragments are large, the expense of collisions dictates that more TX_OPs be allotted per subset to avoid collisions. The access mechanism operates such that each node that belongs to the transmitting subset randomly chooses one TX_OP out of the TX_OPs allotted for the subset. When the allotted TX_OPs are over, a new subset is chosen by traversing the ring in the 'Nxt_indx' direction. Figure 9 shows the ring traversal process. The choice of a new subset is done as shown with the CSR becoming PSR, and the first estimated backlogged node becoming the CSR.

With the access rule as described above, TX_OPs may result in valid transmissions, blank slots (TX_OPs), or collisions. The next subsection describes the state update mechanism for each type of MAC activity. The rest of this subsection describes the collision resolution process and the role of dummy record in state ring traversal.

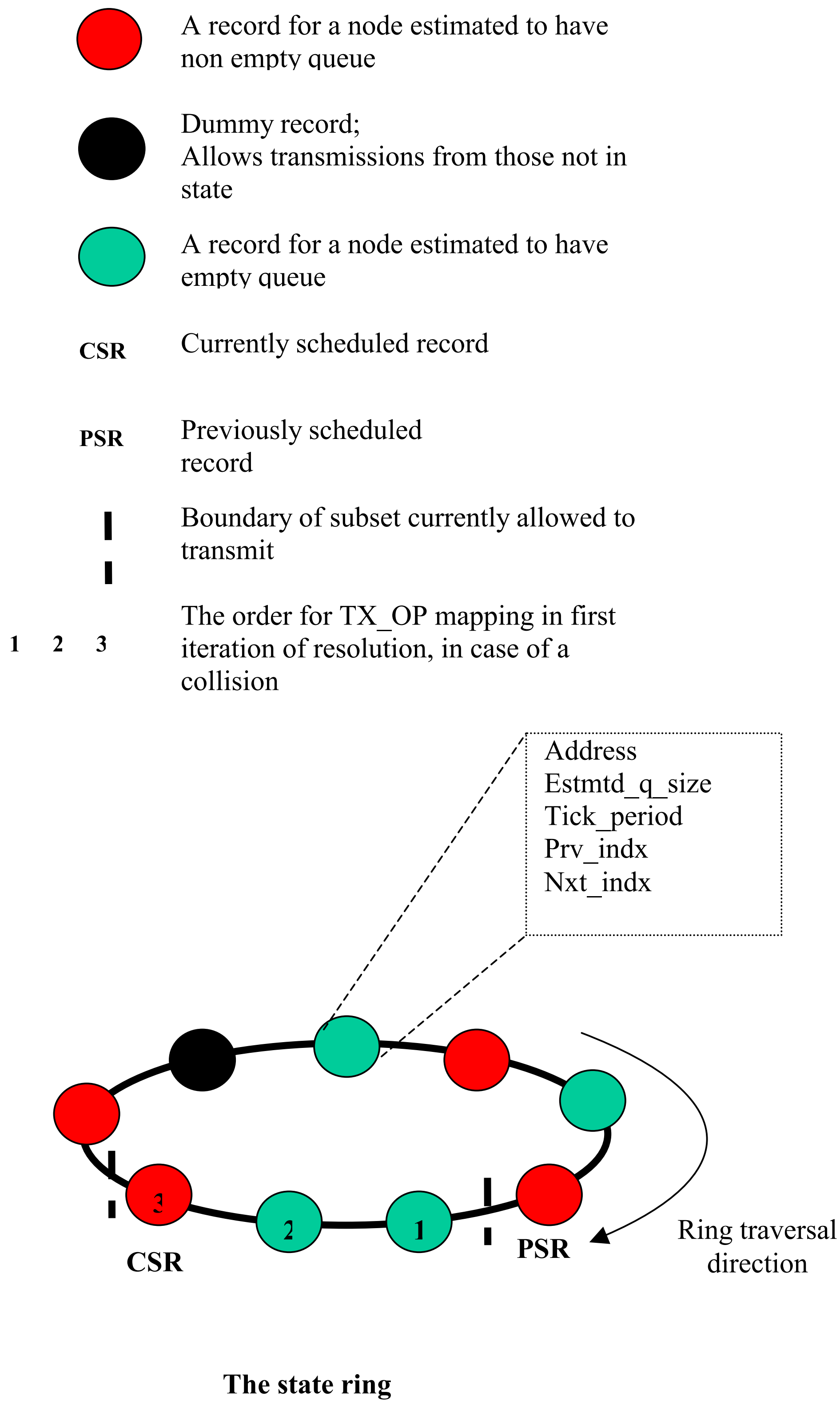


Figure 8. The state-keeping in CSMA/ISS

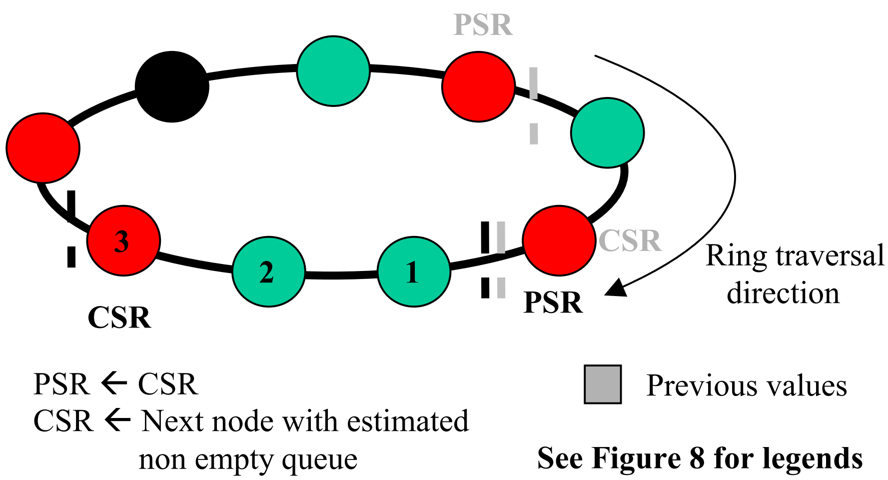


Figure 9. The state ring traversal

If a TX_OP results in a collision, the resolution process begins immediately, new TX_OPs are spawned for collision resolution, and the ring is not traversed until the collision is resolved. Thus, the collision resolution mechanism is expected to be as close to a closed tree based mechanism as possible. Hybrid mechanisms discussed in chapter 4 may also be employed. The exact method of implementation of a resolution mechanism is channel dependent, and is not a part of the framework specification. The only requirement by the framework is that the mechanism be such that the end of a resolution cycle is reliably inferred by all nodes through a specific rule. This is so that all nodes traverse the ring together. That is, the resolution mechanism should be designed to be a *closed* one. Note that since the subset of nodes that are allowed transmission is known, the collision set is known too. This knowledge of the collision set helps in implementing a closed collision resolution mechanism, even if the mechanism employed is not tree splitting based. To summarize, a collision immediately spawns new TX_OPs for resolution, and the state ring is traversed once the new TX_OPs and the allotted TX_OPs for the subset are over. The collision resolution mechanism is required to be such that it is 'closed', and the end of additional TX_OPs spawned for resolution is known. Chapter 7 describes a closed tree mechanism employed in our example implementation.

Determinism in the first step of collision resolution

As mentioned in chapter 5, the order of records in the state ring aids in the fast resolution of any collisions that may happen. In the first round of resolution, any collision resolution mechanism that is employed would involve choosing a random TX_OP from a given number of TX_OPs that are spawned. Revisiting Figure 8 and Figure 9, we observe that the nodes in the subset between PSR and CSR are consecutively numbered. This numbering is used so that the choice of TX_OPs in the first round of resolution is deterministic instead of random. Any node that has collided chooses a TX_OP numbered the same as its number in the subset. Since all nodes in a subset have unique numbers, they choose different TX_OPs, and thus may resolve a collision in the first round itself. This is the basis of deterministic fast collision resolution in CSMA/ISS. In case, the number of a collided node in the subset is more than the number of TX_OPs spawned by the resolution mechanism, it chooses a random TX_OP. Also, if the collision is not resolved in the first round, subsequent rounds are completely based on random choices of TX_OPs. Thus, CSMA/ISS resorts to the classical method if the deterministic method is unsuccessful in the first iteration.

The role of a dummy record is also significant in the access mechanism tied to ring traversal. Dummy record is treated in ring traversal in the same manner as a node estimated to be backlogged. There is no state-keeping and update of the dummy record, and it is always assumed to be backlogged. When the access mechanism allows nodes from a subset of records that includes a dummy record in it, all nodes that are not in the state ring are allowed to transmit. The number of nodes that are not in state and are backlogged may vary from *zero* most of the time to more than 3 or 4 at times. Since a collision is more likely when the CSR is a dummy record, and a number of nodes outside state are backlogged, the operation of CSMA/ISS may be defined differently for efficiency. If collisions are not very expensive, an initial collision may be allowed in the access process. However, if a collision fragment is of the order of packet lengths, collisions are better avoided by beginning the resolution process before a collision. The best solution is case dependent and is left open in CSMA/ISS. One method to achieve this is to allow all backlogged nodes to assert a slot signal (short duration) and begin a collision resolution cycle. If no signal is asserted, the ring traversal process can advance, otherwise a collision resolution cycle begins without any overhead of a long collision fragment. The important idea here is that dummy record allows for nodes to transmit and enter a state ring. Subsequently, they may transmit as the access mechanism traverses the state ring through their records.

6.2.3 State update per traffic class

The state update process in CSMA/ISS is carried out on valid packet transmissions, and during the transition from one subset of nodes to another in ring traversal (i.e. at the end of the set of TX_OPs allotted for a particular subset of nodes to transmit).

When a packet is transmitted, the broadcast nature of the medium causes all nodes to receive the packet (assuming full connectivity). On reception, CSMA/ISS aims to update the Tick_period and the Estmtd_q_size fields of the state record for the transmitting node. In simple terms, the estimated queue size and the arrival rate for the node is updated based on a binary piggybacked feedback (as described in 5.5.2) in the header. On every packet reception, the following rules are followed.

1. If the transmitting node is not in the state ring, a new state record is created for it and inserted *previous* to the CSR. The position is chosen just before CSR to put the transmitter in the end of the list of active nodes beginning at CSR. This helps in fairness, as the most recent transmitting node would access the channel last in the current list. Based on the binary feedback received in the header, the initial Tick_period and Estmtd_q_size fields are chosen from two initial default values for each. The initial Estmtd_q_size is chosen as either 0 or 1. The initial Tick_period is chosen such that the initial estimated queue-filling rate is high enough so that any initial error in estimation is such that the estimated queues are non-empty rather than empty. Thus errors cause short blank slots instead of longer collisions.

   In case the state ring is full with the maximum number of state records, no new record is created, and no state kept for the transmitting node. All nodes that cannot be included in the state ring may still transmit at the dummy record's transmission turn. The channel access process during such transmissions is just the classical collision resolution mechanism in operation. For this reason, the maximum number of state records kept (MAX_STT_KEPT) must be enough to accommodate all active nodes, a large percentage of time. Section 6.6 discusses the implications on storage required.

2. If the transmitting node is in the state ring but is not the node corresponding to the CSR, the node's record is moved to be the previous record to CSR. The logic, as in 1 above, is to put the most recent transmitting node at the end of the list. While this improves fairness, it also allows for a correction mechanism in state-keeping. The

state rings at various nodes may become different from each other due to random errors. The placement of the state record for the most recently transmitting node to the position previous to CSR brings concurrency back to portions of the state ring. Continuous such operation leads to continuous correction, and prevents loss in performance if different subsets are inferred to have access rights by different nodes.

3. If the transmitting node is in the ring and is the node at the CSR, the position of the record is not changed. Note that the record placement policy as outlined in the numbered list here works to automatically correct the positions of records in the state ring. For example, all nodes place the record of the most recently transmitting node to the 'end' of the list either as the CSR, or the node previous to CSR.

Besides the positions of state records, the estimated queue size and arrival rates are also updated on packet receptions as follows.

The update function

The queue size and arrival rate tracking function in CSMA/ISS involves beginning the estimates at a default value, and incrementing or decrementing them according to the feedback received in every packet. The single bit feedback in a packet header denotes the following.

- 1: a feedback of 1 implies that the real queue size of the node is greater than its Estmtd_q_size field in the state ring. On receiving such feedback, all nodes in the network do not decrement the Estmtd_q_size field for the transmission, and decrease the Tick_period field to increase the estimated arrival rate.
- 0: a feedback of 0 implies that the real queue size of the node is less than or equal to its Estmtd_q_size field in the state ring. On receiving such feedback, all nodes

decrement the Estmtd_q_size field of the transmitting node by 1 to allow for the transmitted packet. They also increase the value of the Tick_period field for the node, so as decrease its estimated filling rate.

The main variable that is updated on packet reception is Tick_period. CSMA/ISS design dictates that the Tick_period value be maintained between a minimum (MIN) and a maximum (MAX) value chosen depending on the minimum and maximum values of the arrival rate expected on the network. The MIN value depends on the maximum arrival rate expected at any node. This may be chosen as the time taken to transmit a small (say a 100 byte) packet on the shared channel. Thus, it may depend on the channel rate. The MAX value depends on the maximum inter-packet arrival time such that any larger inter packet spacing is considered two different flows. Thus it may be chosen as the maximum inter packet interval so that the packets are still considered part of the same flow. As an example a possible value can be the time taken to transmit a fixed number of MTU bytes packets. The fixed number may be the number of priorities times the mean number of active nodes expected in the network at any time. These numbers are just some suggested values, and need not be the ones chosen. While the Tick_period field is to be varied between the chosen MAX and MIN values, the functions employed to increase and decrease the value at each step can be more flexible. There were no experiments performed to investigate ideal functions for the job. Heuristically, the functions should be such that closer the current value is to one of the extremes (MIN or MAX), the smaller are the changes to the value in that direction, and larger are the changes to the value in the opposite direction. One suggested set of increase and decrease functions, as shown in Figure 10, are X^2 and X(2-X) for MIN and MAX values normalized between 0 and 1.

This is the set of functions employed in the example implementation discussed in chapter 7. X^2 is the decrease function, and $X(2-X)$ is the increase function. An important variable in the implementation of updates of Tick_period is the step size DELTA that is used in the increase and decrease functions mentioned above. Smaller values of DELTA provide finer granularity in tracking, but may increase the number of packets (i.e. the number of times an update is done) required to be transmitted for the estimates to resemble the actual arrival rates. The increase or decrease updates work in the following manner. The current value is first changed by DELTA in whichever direction (increase or decrease) the feedback suggests. This value is used as the input to either the increase of the decrease function, and the result is the new value. The choice of DELTA is also heuristic to allow for a good number of steps in traversing monotonously from one extreme (MIN or MAX) to another. Note that the number of steps mentioned here is not the total number of possible values the variable Tick_period might get. The total number of values Tick_period may get assigned in the process is significantly larger than the number of monotonous steps, as shifts from one function to another (i.e. from increase to decrease or vice versa) create a number of intermediate values that may get assigned. The values depend on the sequence of 1 and 0 feedback received for the variable.

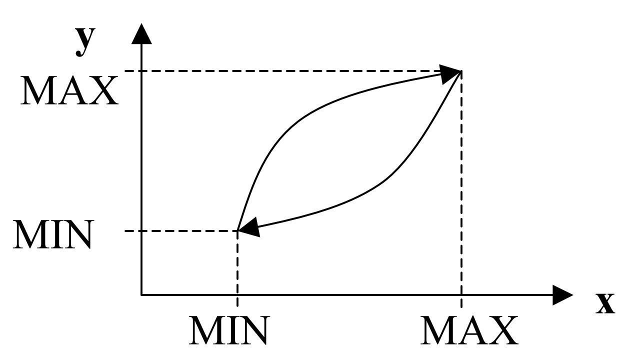


Figure 10. An example of increase and decrease function for Tick_period

Estmtd_q_size, the other variable is updated as follows.

- Incremented by 1: It is incremented by one every Tick_period amount of time. Since the Tick_period varies, so does the frequency with which the variable is incremented.
- Decremented by 1: It is decremented by one after a transmission by the source node if the feedback bit is 0. It is not decremented if the feedback bit is 1 because the feedback implies the real queue size is larger than the current estimated queue size.
- Assigned the value 0: During the transition from one subset of nodes to another in ring traversal, it is known whether the CSR transmitted or not. If the CSR did not transmit during the TX_OPs for the subset, its real queue is empty. Thus the estimated queue size for CSR is also assigned the value 0. Whenever the Estmtd_q_size variable is reset to 0, the access mechanism begins the process to check if the source node has become inactive. This is discussed next.

6.2.4 Removal of records from state ring

When the nodes with records in the state ring become inactive, the corresponding records need to be taken out of the ring. This is so that new active nodes may use the limited number of state records, and so that the state ring is maintained as small as possible for low latencies and low processing overhead. The first step towards a node being inactive is that its queue size goes to zero. If the queue size stays zero for a long enough time, the node may be considered inactive. This is the method employed for estimating nodes as inactive and removing their records from the ring in CSMA/ISS.

The process of removing records proceeds as follows. The Estmtd_q_size field for a record is reset either due to no transmission when the record is the CSR, or through a transmission by the source node, so that the filed is decremented to 0. Whenever the field is reset, a *dormancy* timeout is begun. If there is no transmission by the source node for the timeout time, the record is removed from the state ring. The virtual arrival process through which the Estmtd_q_size field is incremented by 1 every Tick_period continues even when the dormancy timeout is set. Thus, the field may become non-zero, and may cause the record to become CSR and get access to the channel. If the source node transmits before the timeout, the timeout is aborted, and the estimation process continues normally. However, if the source node becomes CSR and still does not transmit while the timeout is on, the Tick_period increase function may be different from the normal one. For example, the Tick_period may be doubled at every such instance. This causes the virtual filling rate to reduce faster than through the normal function, and reduce the extra TX_OPs wasted on a node that may have become inactive. The choice again is channel characteristic dependent. If collisions are expensive, they can be avoided in favor of shorter blank slots of unutilized TX_OPs by not decreasing the virtual arrival rate too fast. Besides the dormancy timeout, another method of removing records from the ring is possible. Once the Tick_period period reaches the MAX value, the virtual inter-packet arrival time is of the order of the time for inactivity. Thus, if the Tick_period reaches the MAX value, the corresponding node may be considered inactive, and its record may be removed from the ring.

This section described the MAC operation of individual priority classes in CSMA/ISS. The next section describes the overall operation of CSMA/ISS.

6.3 Multi-priority overall operation

Fair access over multiple priority classes in CSMA/ISS is achieved in a manner similar to the fair access achieved *within* each priority class. As mentioned in chapter 5, a weighted-fair serve-and-pause type of access mechanism is proposed in CSMA/ISS. The following subsections describe the operation is detail.

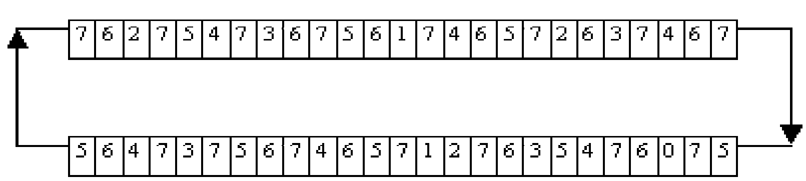
6.3.1 The traffic class ring and hierarchical fairness

CSMA/ISS is designed to achieve hierarchical fairness, with two levels of fairness. At the top level, fairness across 8 priority classes in the network is expected to be according to *static* weights for each class. As we observe below, changing of weights of the different classes in CSMA/ISS is possible only with significant effort. The design is aimed at static and distinct weights for 8 different classes. At the second level of fairness, all nodes transmitting in a particular class are expected to achieve equal channel access. Section 6.2 above described the MAC operation of CSMA/ISS for a single class. Fairness within a class is achieved by round robin access to all active nodes, with each access allowing approximately an MTU amount of data. In order to implement the fairness across priority classes, a similar serve-and-pause mechanism is proposed for access to the MAC operation of each class.

The serve-and-pause access in a round robin manner that implements weighted fairness in CSMA/ISS is achieved through traversing a ring similar to the state ring. This ring, referred to as *the traffic class ring* or simply *the class ring*, is a static data structure as shown in Figure 11. The class ring has a fixed number of records arranged as a closed looped list. Each record comprises of the following two fields.

1. **Class number**: This denotes the priority class the record represents. As the ring is traversed, the MAC operation of the class corresponding to each record is invoked to allow transfer of a fixed amount of data (for example MTU bytes). Once this data transfer is achieved, the class ring is traversed and MAC operations of classes corresponding to other records are invoked.
2. **Active-inactive flag**: This is a 1-bit (or binary) flag that denotes whether the priority class represented by this record is active or inactive. Here, active implies that the MAC state for the class is set up. Equivalently, if there is any traffic in a particular priority class, the state ring for the class is set up, and the class may be referred to as 'active'. This flag aids in skipping the records for inactive classes in the class ring traversal.

Fairness according to weights is achieved with the class ring traversal by having the *weight* number of records for each class in the ring. Thus, the MAC operation for each class is invoked its weight number of times in one round. A fixed amount of service per MAC invocation causes the different classes to receive channel access according to their weights. As an example, Figure 11 shows a class ring with classes 7 to 0 with weights 15,11,8,6,4,3,2, and 1 respectively. To improve fairness, the records for different classes should be uniformly interspersed among each other, as shown in Figure 11.



Weights for priority classes 7 to 0 = 15,11,8,6,4,3,2,1

Figure 11. An example traffic class ring with a class number for each record

Figure 12 shows an abstract visualization of the overall CSMA/ISS operation. At the top level, CSMA/ISS traverses the class ring over all active records in a particular direction, as shown. Each record in the class ring may be identified by an index called its *position*. The class ring traverses all positions that are flagged active, and invokes the state ring for the class at that position. The state ring, when invoked, is traversed as described in section 6.2 until MTU amount of data is transmitted in the class, or until there is no backlog in that class. The class ring then moves to the next active position and invokes the corresponding state ring in the same manner. We observe that hierarchical fairness is achieved through such hierarchical ring traversal. The class ring traversal achieves weighted fairness across classes, and the traversal of state rings for each class achieves fairness within classes. The next subsection details the traversal mechanism with some explicit position information in packets.

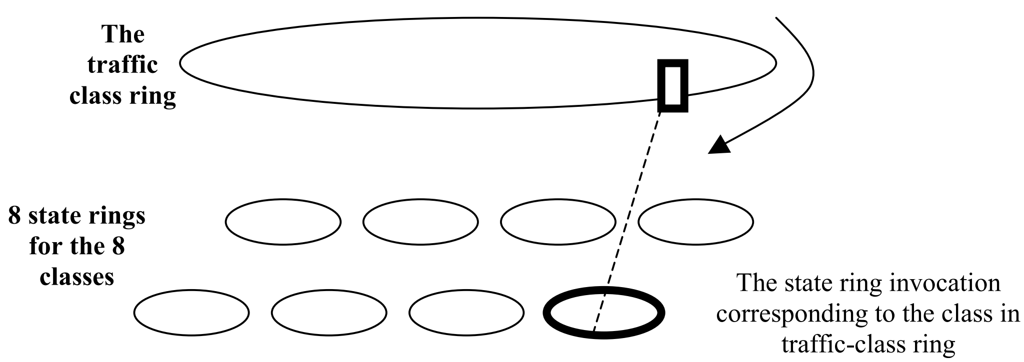


Figure 12. A visualization of CSMA/ISS overall operation with the class and state rings

6.3.2 Explicit feedback: position and relinquish information

Consider the overall header overhead in data packets that is required for CSMA/ISS. A single bit overhead per class is required for tracking the queues and arrival rates of active nodes as described in 6.2. For such explicit feedback information for all the eight classes, the net header overhead required is 8 bits, or a byte. Let this byte be referred to as the *feedback byte*. As we describe below, an additional byte of overhead may be employed per packet to efficiently implement the ring traversals in the access mechanism.

Consider an extra header byte, called the *position byte*, with the following two fields.

1. *Position* (6 bits): This field refers to the position in the class ring at which the transmitter is in its class ring traversal. Since the class ring is a static data structure set at all nodes, a position can be mapped to a priority class. In CSMA/ISS, the class information for a packet is transmitted in the form of the position in the class ring. This explicit transmission of the position information allows the network to remain synchronized with respect to the position being traversed in the class ring. All nodes, on the receipt of the position field in a packet header, set their current positions in the class ring to the same. This is regardless of the ring position they were at before the transmission. The total number of positions with a 6-bit field is 64. Since the number of positions in the class ring for each class add up to their fairness weights, the sum of weights of all classes may add up to 64 maximum. The example class ring shown in Figure 11 involves 50 total positions, and is quite representative of the expected weights for various classes. The weight distribution, however, is a design choice. Should a weight distribution be required such that the total weights add up to more than 64, one additional header byte may be employed. In any case, the net header overhead due to CSMA/ISS is minimal.

2. *Relinquish bits* (*R1 and R0*; 1 bit each): The relinquish bits aid in the traversal of the class and state rings maintaining fairness. R1 denotes if the position in the class ring is relinquished, and R0 denotes if the transmitting node is relinquishing the right over the channel. They are employed to allow nodes or traffic classes to *capture* the right to channel access until they transmit MTU amount of data at a class ring position. If the position in the class ring is relinquished, it implies that the next transmission is expected from the class corresponding to the next active position in the ring. In such a case the other relinquish bit is irrelevant. However, in case the position in the class ring is not relinquished, the transmitting node may or may not relinquish its right over the channel. If the transmitter relinquishes the right, the state ring for the same class is traversed, and a new subset is allowed transmission. The CSR in the new subset may relinquish the access right, or the class position then. In case a node does not relinquish the access right over the channel, the network expects more packets by the same node until it asserts its relinquish bit (R0). Nodes in CSMA/ISS are expected to keep track of the amount of data transmitted with the protocol being at one position in the class ring. For fairness, the idea is to allow MTU amount of data for the class corresponding to the position. This may be accomplished in multiple ways, and CSMA/ISS does not define the method specifically. The suggested method is to allow one node to keep transmitting until it transmits MTU bytes or has no backlog. If the node transmits less than MTU bytes, and has no backlog, it should relinquish only its access right, and should not relinquish the class position. This allows the next subset in the state ring of the class to transmit if it

has packets. To summarize, R1 relinquishes the current class ring position, and allows traversal of the class ring; and R0 relinquishes a node's access right to allow traversal of the state ring. They are employed to allow fixed amount of data to be transmitted per class ring position.

Note that the sizes of packets at nodes may be such that the transmission or not of a single packet may cause the net data transmitted for a position to be significantly either below or above MTU. For fairness, such packets may be probabilistically transmitted or not, as described in 5.5.3.

Summarizing, an extra byte of header overhead in CSMA/ISS aids in achieving a synchronized MAC across the network, and in implementing fine-grained fairness.

6.3.3 Setting positions to be active or inactive

An important constituent of the framework is the method of setting positions in the class ring to be active or inactive, and the process of *traversing* the class ring. The relinquish bit R1 is employed to signal the network that the next active position in the class ring be used to invoke channel access. However, once such a packet is transmitted, the TX_OPs for the MAC operation of the new (possibly same even, depending on the position) class may begin immediately, or the protocol may allow for some transient short blank slots. The transient TX_OPs may be employed by nodes to transmit data for an inactive position between the two active positions. Thus, a position may become active with the first transmission by a node in the class corresponding to its position. Such transmissions cause all the positions with the particular class number to become active. Positions in the class ring may become inactive if the state ring for the corresponding class becomes empty. This happens as the records in the state ring are gradually removed from the ring as described in 6.2.4.

Besides on explicit notification through the R1 bit, the traffic class ring may also be traversed if there are no transmissions in the class corresponding to the current position. The framework does not specify a particular rule for such traversal. However, a fixed number of unused TX_OPs may be sufficient to indicate such traversal. This portion of the framework may thus be adapted to specific channels in specific ways.

Sections 6.1 to 6.3 above provide a detailed description of the CSMA/ISS framework. The rest of this chapter discusses various aspects and properties of the framework described above.

6.4 Error handling and resynchronization

The implicit scheduling approach in CSMA/ISS depends on identical state-keeping at all nodes for good performance. The state at different nodes may become dissimilar if they perceive different channel activity. CSMA/ISS is designed with mechanisms for automatic state correction/synchronization to maintain performance in such cases. There are two main types of errors that affect the framework performance.

1. Bit errors: If some nodes receive a packet with bit errors, they cannot read the data in the packet. If some nodes can read the data while others cannot, they may update their state information differently. This can cause dissimilarities in the states kept at different nodes, and possibly some loss in performance. This type of error does not affect the framework performance significantly if all nodes can at least interpret the durations and type of channel activity identically. CSMA/ISS recommends making the ‘most likely’ assumptions in such a case, and continue the mechanism. The

dissimilarities in the state happening in such a case are minor, and may not affect the performance significantly. Resynchronization mechanisms described below correct some of the dissimilarities, thus incurred.

2. Connectivity loss: Random loss of connectivity in the network is an error that can cause significant dissimilarities in the MAC state of different nodes. Loss of connectivity implies that some nodes that are part of the network do not receive transmissions by some other nodes. Since CSMA/ISS incorporates feedback through received channel activity along with explicit feedback, nodes may update their states differently depending on whether they receive the transmissions or not. The error condition is also commonly known as the hidden node problem. Specific receiver based mechanisms that employ request-to-send (RTS) and clear-to-send (CTS) packets have been designed to address the issue [41]. Multi-transceiver wireless systems, in which a receiver asserts an out of band narrow channel whenever it is receiving, are also proposed as a solution [42]. These approaches may be employed along with CSMA/ISS to improve performance in hidden node scenarios. Since CSMA/ISS involves both implicit and explicit feedback, more explicit feedback may be employed in environments in which the hidden node problem is more likely. The next section explores this is some more detail. Below are listed the inherent mechanisms in CSMA/ISS that work to maintain state synchronization among different nodes.

Performance optimization mechanisms

1. Continue with *most likely* assumptions: In case of any doubt while in operation, the framework recommends that the protocol make assumptions as to the most likely

scenario, and proceed. For example, if the duration of a particular channel activity is perceived as within the limits of a valid packet duration, but the data is not read, the protocol may assume the transmitter is the CSR, and proceed with ring traversal. The relinquish bits may also be assumed based on the duration of the activity. For example, if the duration is close to that required for MTU bytes, the relinquish bits may be assumed *set* in the packet.

2. State-ring re-ordering: In the state kept for each class in CSMA/ISS, the order of records in the state-ring is the most important information. This is so because the subsets of nodes allowed transmission by the access rule, and the initial collision resolution order depend on this order. In order to maintain a common order of records in the state rings at different nodes, CSMA/ISS involves the record re-arrangement mechanism for records of transmitting nodes. As described in 6.2.3, the record for a node that transmits a packet is placed as *previous* to the CSR, if the record is not the CSR itself. This rule places the record logically to the end of the list of nodes allowed transmission, beginning from *next* to CSR. This happens for all nodes, and thus portions of the state ring are continuously synchronized across the network.
3. Synchronization with explicit feedback: Explicit feedback is the best form of synchronization, as it involves no ambiguity in interpretation. Explicit synchronization is defined in the base definition of CSMA/ISS in the form of the *position byte* in packet headers. The position field in the byte explicitly declares the current position at which the transmitter's class ring is executing. All receiver nodes are expected to set their class ring execution position to the same. The relinquish bits also provide explicit feedback as to when to traverse to the next records in the class

ring and the state ring. In case nodes lose synchronization with respect to the position in the class ring, they can simply wait till the next valid transmission and set their MAC execution from the position declared in the packet's header.

Another error scenario that is resolved efficiently due to the explicit position information is the occurrence of collisions between packets of different classes. Different nodes may begin their resolution process under their own assumptions of the current class of operation. The first valid transmission in the collision resolution process however would separate the interaction between MAC operations of different classes. The position byte in the packet would cause all nodes to synchronize with respect to the position (or equivalently the class) in the class ring.

While the explicit feedback defined in the base definition of CSMA/ISS may be enough for a good performance in reliable fully connected channels, extra such feedback may be employed in other environments. For example, CSMA/ISS may be used with RTS/CTS packets to avoid hidden node problems, and the RTS/CTS packets may also carry the header bytes defined in the base definition of CSMA/ISS. Thus states may be synchronized in hidden node environments too. Concise explicit feedback may also be defined in imaginative new ways to adapt CSMA/ISS to different environments.

6.5 Adapting to channel characteristics

There are two main channel characteristics of a CSMA channel that affect the manner in which CSMA/ISS is adapted to the channel. The first is whether collisions can be detected or not, and the second is the nature of full connectivity in broadcast segments.

If collisions are not detected in a CSMA network, the collision fragments are of the order of the largest packet in the collisions. This implies the capacity loss in case of a collision is much larger compared to that in case of an unused TX_OP (which is a short blank slot in most CSMA environments). In such a scenario, CSMA/ISS should be designed to avoid collisions as much as possible. This may be achieved through multiple measures. The number of TX_OPs that are allowed for a subset of nodes from the state ring may be increased. The collision resolution mechanism may also be chosen to operate with a higher number of TX_OPs per round of resolution. The transition from one position in the class ring to the next may be designed to have some guard TX_OPs. This allows intermediate positions to possibly become active by allowing transmissions in their priority class without colliding. Another measure that biases CSMA/ISS towards unused TX_OPs in place of possible collisions involves the queue and arrival rate tracking mechanism. The increase and decrease function may be chosen such that the virtual queue filling rate is biased towards keeping the Estmtd_q_size variable greater than 0. The above listed mechanisms emphasize how CSMA/ISS is coupled with classical MAC mechanisms that can be tuned to suit the channel requirements.

The loss of MAC performance in CSMA networks due to random loss in connectivity is well known [41]. Some receiver-based mechanisms such as employing RTS/CTS and busy tone signaling are proposed as solutions to the problem. Such mechanisms may be employed in conjunction with CSMA/ISS too. The use of CSMA/ISS explicit feedback in RTS/CTS was alluded to in the previous subsection. Note that CSMA/ISS employs both implicit feedback based on the MAC activity in the carrier-sensing channel, and explicit feedback piggybacked in data or control packets. In case the connectivity in the channel becomes unreliable, so does the feedback information that may be associated with channel activity. Thus if connectivity is unreliable, a blank channel may not be reliably assumed as 'no transmission' in the network. Since the performance gains from implicit feedback are reduced, more explicit feedback may be required in order to maintain optimal performance in CSMA/ISS. However, explicit feedback involves some extra channel expense. Thus, the performance of CSMA/ISS is expected to be the best in reliable fully connected CSMA channels. As the connectivity is reduced, the performance gains with CSMA/ISS are expected to be reduced too. Thus, the performance gains of CSMA/ISS in wireless networks are expected to be less than those in wired collision detecting networks. However, theoretically, a CSMA/ISS MAC protocol may always be designed to perform better than or equal to one that only employs classical mechanisms. This is so because state-keeping implies the protocol has some extra information, and using the information for performance gains is entirely design dependent. As the connectivity of the channel reduces, CSMA/ISS may be adapted to employ more classical methods of access that are particularly designed for specific channel environments.

6.6 Storage overhead

CSMA/ISS, with its comprehensive state-keeping at all nodes in the network requires more memory to operate than most classical MAC mechanisms. In this subsection, we quantify the storage overhead.

The state record is the basic unit of extra storage required in CSMA/ISS. The fields of MAC address (6 bytes), Estmtd_q_size (2 bytes), Tick_period (4 bytes), and the two pointers (4 bytes each) require 20 bytes of storage in total. Let the maximum number of records kept per priority be N. The total number of records required with 8 classes is 8N, leading to the net requirement of 20 time 8N, or 160N bytes. The storage required for the other variables employed in CSMA/ISS is significantly smaller compared to the storage required for state records. For each class, a few state variables are required to maintain the state of the MAC operation for the class. For example variables are required to store the backoff information for collision resolution. Let the storage required for such variables be C bytes. For 8 classes the net storage required is 8C bytes. C can be expected to be of the order of a few tens of bytes. Assuming C to be 20 bytes, the storage overhead of 8C results in 160 bytes of overhead. Adding this overhead to the 160N bytes of overhead required for state records, the net overhead amounts to 160(N+1) bytes. The traffic class ring also requires a fixed amount of storage. The storage required per record is less than a byte. Three bits are required to store the class number, and 1 bit to store the active-inactive flag. Assuming 1 byte of storage required per record, a maximum of 64 records in the class ring requires 64 bytes of overhead. This overhead is again significantly smaller than the overhead required for state records. Thus, the net storage required in CSMA/ISS is of the order of 160(N+1) bytes. The maximum number of record kept per state ring (N) is an important protocol parameter. It should be high enough so that the probability of saturation of the state ring is extremely small. A value of 30 to 50 is representative of the maximum number of active nodes supported in a broadcast segment. Assuming these values for N, the net overhead required is of the order of 4800 to 8000 bytes. This is equivalent to 3 to 5 MTU sized packets in Ethernet. We observe that the storage overhead in CSMA/ISS is moderate, and does not present any implementation challenges.

6.7 Processing overhead

The operation of CSMA/ISS, as described in the previous sections of this chapter, involves a processing overhead larger than most classical mechanisms. In this subsection, we quantify this overhead. We observe that the overhead is moderate and conducive to hardware implementation.

The processing overhead in CSMA/ISS may be classified into the following types of operations.

1. Copy or assignment operations: A number of copy and assignment operations are required in CSMA/ISS as the mechanism proceeds. Timeouts values that are stored in variables are copied to other variables that are decremented every clock cycle to implement the timeout. For example, the Tick_period value is copied and decremented to implement a timeout after which the Estmtd_q_size variable is incremented. A constant but possibly different number of such operations are required at various steps in the operation of the access mechanism. Since copying in physical memory is extremely lightweight operation, the processing overhead for such operations is insignificant.
2. Increment/Decrement operations: A majority of the extra operations required in CSMA/ISS is of increment-decrement type. A number of variables involve operations in which they are incremented or decremented by 1. All timeouts and time counting operations involve decrementing an integer by 1 every clock cycle. The Estmtd_q_size field and backoff counters are decremented as the MAC operation proceeds. The Estmtd_q_size field is also incremented every Tick_period. Tick_period counting is itself a continuous decrement operation. The

increment/decrement operations are also extremely lightweight and conducive to hardware implementation. Thus, the processing overhead involved in such operations is also small.

3. Random number generation operations: Similar to most classical MAC protocols, CSMA/ISS also involves random number generation for collision resolution. Such number generation may also be employed for making various probabilistic decisions. Overall, the overhead is small, and similar to classical approaches.
4. Add and shift operations: Some add and shift operations are required in CSMA/ISS for every transmission. For example, all nodes keep track of the number of bytes transmitted at a position in the class ring. This is required so that nodes transmit approximately MTU bytes per position. Packet sizes are required to be added for such operation. Another operation that requires add-and-shift operations (or equivalently adds and a multiply operation) is the computing of a new Tick_period after every packet reception. Since the new value is one of only two possibilities given the old Tick_period, it may be pre-computed. The values may also be stored as a table, so that the only overhead required is that of table look up. Even if the computation is performed in real time as protocol progresses, this operation involves a single multiply and possibly two additions. Thus, although the processing required is more than other operations involved in CSMA/ISS, it is still low. The processing is again conducive to optimized hardware implementation.
5. XOR operations: On every packet reception, CSMA/ISS involves a lookup of the source address in the state ring of the appropriate class. Such a lookup may be implemented as a series of exclusive-or (XOR) operations. The source address of a

received packet can be ‘XORed’ with the addresses in the records of the corresponding state ring. If the operation results in a 0 value, the address is found. The number of XOR operations required is the number of records in the state ring. In extreme network conditions, this number may be in the tens or twenties. The operations are however extremely lightweight and conducive to hardware optimizations. Thus, the net overhead due to XOR operation is not expected to be significant in CSMA/ISS.

From the list above, we observe that the processing overhead required to implement CSMA/ISS is moderate and extremely conducive to optimized hardware implementation.

In this chapter, we described the proposed CSMA/ISS framework in detail. CSMA/ISS, as mentioned earlier, is a framework that can be adapted to different CSMA MAC environments to implement actual protocols. Next, we present an example implementation in a wired environment, and its performance evaluation results.

CHAPTER 7

EXAMPLE IMPLEMENTATION FOR WIRED NETWORKS

In order to evaluate the performance of CSMA/ISS, it was adapted into a MAC protocol for a wired fully connected CSMA network. This chapter describes the design decisions that define the protocol beyond the framework definition.

7.1 A fully connected wired CSMA channel

The example implementation of CSMA/ISS as a MAC protocol was aimed at a reliable fully connected wired CSMA channel. For example, the protocol may be employed for Home PNA networks, or a broadcast network over unshielded twisted pair (UTP) wires. The network is assumed to detect collisions, so that collision fragments are small compared to valid packet transmissions. The channel activity in the protocol may be one of the following types.

1. Blank slot: A short duration for a slot is defined. If a TX_OP is not used for transmission by any node, it is considered over in a short duration of SLOT_SIZE. Such an unused TX_OP is a blank slot.
2. Slot signal assertion: The protocol employs signal assertion as a means of feedback, as described in the next subsection. The duration of assertion is defined as fixed and equal to SLOT_SIZE.
3. Valid packet transmission: Variable sized packets with a maximum size of MTU (1500 bytes) may be transmitted on the medium.
4. Collision fragments: Once a collision is detected over the medium, the transmitters of the colliding packets abort their transmissions after a fixed duration called

COLL_FRAG. The COLL_FRAG duration is required to be greater than the SLOT_SIZE duration. This is so that the slot signals can be distinguished from collision fragments based on their durations. Since there might be random detection delays in CSMA networks, the difference between COLL_FRAG and SLOT_SIZE should be enough to absorb these delays.

5. Inter-frame gap (IFG): All channel *asserting* activities such as slot signals, packet transmissions, and collision fragments are followed by a mandatory fixed duration of channel inactivity called the inter-frame gap (IFG). TX_OPs begin immediately after the end of IFGs.

Any channel activity in the protocol is interpreted as one of the above-mentioned types. The next section lists the parameters and mechanisms chosen to define the protocol.

7.2 Protocol implementation details

CSMA/ISS is parameterized with respect to a number of variables and mechanisms. These parameters chosen to define the example protocol implementation are listed below.

The class ring chosen for the protocol is the example ring shown in Figure 11. The weights for classes 7 to 0 are 15,11,8,6,4,3,2, and 1 respectively. The total number of records in the class ring is thus 50. The records for different classes are interspersed uniformly as shown in Figure 11. This uniform interspersing allows transmissions to begin randomly from any record without affecting fairness.

The protocol is designed for 100 megabits per second (Mbps) channel rate. With this data rate, the protocol parameters are chosen so that they are comparable to 100 Mbps Ethernet parameters. The collision resolution mechanisms of Ethernet and the CSMA/ISS example protocol are different. Therefore, the collision fragments and slot sizes in the example implementation may not be chosen exactly the same as in Ethernet. Moreover, as mentioned in 7.1, the COLL_FRAG duration is required to be larger than the SLOT_SIZE duration in the example implementation. The parameters were therefore chosen so as to satisfy the requirements of a viable design, and also so that the performance of the two collision resolution mechanisms is identical for 16 node operation. This may be observed in 8.4, and in Figure 22. The parameters values chosen for the example implementation are as follows.

1. SLOT_SIZE: 0.64 microseconds
2. IFG: 0.96 microseconds (same as Ethernet)
3. COLL_FRAG: 1.6 microseconds

Another important design parameter in CSMA/ISS is the number of TX_OPs allotted for every subset of nodes in the state ring that are allowed to transmit together. In the example implementation, the number chosen is one. That is, all nodes in a subset from the PSR (excluding) to the CSR (including) are required to transmit in one TX_OP assigned for the set. Thus, if multiple nodes in the set are backlogged, collision results. Since collisions are detected in the network, and the collision fragments are small, such aggressive design performs well. Another important reason for the aggressive design is that the channel is assumed to be fully connected with high reliability. Thus, the probability of different nodes having different states is small enough to not affect the performance. Full connectivity and reliability also imply that the queue size and arrival rate estimates through CSMA/ISS are sufficiently representative of the actual conditions at nodes. Thus, the collision probability is expected to be low even with the aggressive design.

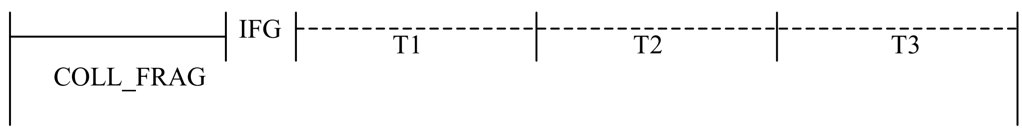
A ternary tree based closed collision resolution mechanism

Collisions in the example implementation are resolved using a closed tree based resolution mechanism. The resolution mechanism is similar to the Home PNA 2.0 mechanism but without the slot signal assertions [21]. It is the mechanism analyzed in [37], and is described briefly as follows.

1. Any collision is immediately followed by *n* TX_OPs, where *n* is a fixed number known to all nodes. It (*n*) is also the degree of tree splitting.
2. Each colliding node chooses one of the *n* TX_OPs to transmit.
3. A TX_OP chosen by a single node ends in a valid packet transmission.
4. A TX_OP chosen by none of the nodes ends in a short blank slot (SLOT_SIZE duration).
5. If multiple nodes choose the same TX_OP, another collision results. This collision is immediately followed by *n* TX_OPs for its resolution. Thus, the resolution process is inherently recursive.
6. The mechanism is closed, and all nodes can infer the end of the resolution process. The collision resolution (CR) cycle (as defined in chapter 4) may be visualized as being composed of the initial collision fragment (CF), followed by an IFG, followed by *n* collision resolution cycles. Figure 13 shows the scenario for $n = 3$. Blank slots and valid transmissions may also be considered as CR cycles with the collision multiplicity (number of colliding nodes) as 0 and 1 respectively. Note that any real

collision involves multiple packets, and the collision multiplicities are greater than or equal to 2.

7. The tree nature of the resolution mechanism is evident in the recursive nature of the resolution process. The recursion is inherent with the existence of CR cycles within CR cycles.



T1, T2, and T3 are CR cycles corresponding to the spawned TX_OPs

Figure 13. An example collision resolution cycle with $n = 3$

The n chosen for the example implementation is 3. The choice is based on the analysis presented in [37]. While the optimal degree of splitting (DTS) for collision multiplicities greater than 3 is higher than ternary, the chosen DTS is 3, as the collision avoidance and resolution process in CSMA/ISS is aided by determinism. Section 6.2.2 described the fast resolution process aided by an ordered collision set in CSMA/ISS.

Serve and pause operation

The serve-and-pause operation of CSMA/ISS for fairness across different priority classes was described in chapter 6. While the use of relinquish-bits to achieve the operation is well specified in the framework, the class ring traversal if no traffic is transmitted for a position is left open to specific adaptation. In the example implementation, the following rules are employed for class ring traversal in the absence of traffic.

- Four consecutive blank slots imply that there is no backlog in the class corresponding to the current class ring position. Thus, the class ring is traversed to the next active position.
- Since the protocol operation, in the process of state ring traversal and collision resolution, may involve more than 4 consecutive blank slots, a special rule for slot signal assertion is defined. The rule specifies that after 3 consecutive blank slots, all backlogged nodes in the particular class assert a slot signal, and then continue the protocol operation. This prevents 4 consecutive blank slots from occurring if there is any backlog in the class.
- The protocol also specifies that 5 consecutive blank slots imply that there is no backlog in any class in the network. The network transitions to an unsynchronized state from which transmission is allowed as soon as there is any packet arrival.
- In order to prevent the network from transitioning into the unsynchronized state, the protocol specifies that after 4 consecutive blank slots, a signal be asserted in the fifth slot by all nodes that are backlogged in any class. The CSMA/ISS operation continues normally after such assertion. Since 4 consecutive blank slots imply that the class ring traverses to a new active position, the MAC operation for this new class may begin after the signal assertion in the fifth slot.

The rules outlined above cause the example implementation of CSMA/ISS to be aggressively tailored for efficiency in a reliable fully connected channel environment. The entire implementation of the CSMA/ISS framework as a MAC protocol is designed specifically for a reliable fully connected channel. This protocol should not be expected to automatically perform well in other channel environments. As described in chapter 6, the framework can be individually adapted to various CSMA channel types.

CHAPTER 8

PERFORMANCE EVALUATION RESULTS

The protocol described in chapter 7 was modeled and simulated in the OPNET modeler [43] to evaluate the performance of CSMA/ISS. This chapter describes the evaluation results.

8.1 Modeled scenarios

In order to comprehensively and fairly evaluate the performance of CSMA/ISS, a number of different MAC protocol scenarios were modeled, and simulated with identical inputs. Below is a list of the protocols simulated for performance evaluation, and the reasons for their choice. All of the protocols below were modeled over broadcast wired buses.

1. ISS_MAC: This is the example implementation described in chapter 7, and the protocol whose performance is evaluated. The maximum number of state records kept per priority class was chosen to be 30. That is MAX_STT_KEPT (N) = 30.
2. ISS_No_Determinism: This is the ISS_MAC with MAX_STT_KEPT (N) equal to 1. Since this protocol does not keep any state records, its performance is devoid of any gains due to state-keeping. The performance gains in CSMA/ISS due to state-keeping can thus be evaluated by comparing performance against this protocol. The protocol essentially involves the classical ternary tree resolution mechanism as the access mechanism within priority classes. The mechanism for fairness across classes using a traffic class ring is the same in ISS_No_Determinism as in ISS_MAC.

3. IDL_MAC: An ideal scheduling MAC protocol was modeled and simulated in order to evaluate how close the performance of CSMA/ISS approaches the benchmark best performance. Ideal scheduling is defined and described in chapter 2. The scheduling algorithm employed for the modeled protocol is the weighted fair queuing algorithm that schedules according to the *virtual transmission end times* of packets [5]. The weighted fair scheduling was designed to implement hierarchical fairness in medium access. The fairness across priority classes was implemented with the same weights as employed in ISS_MAC and ISS_No_Determinism. The fairness within classes was implemented with equal weights for all nodes.
4. Ethernet: This is the standard CSMA/CD Ethernet MAC protocol [20]. It was used to compare the performance of CSMA/ISS with additional protocols in the single priority class scenario.
5. Ethernet_No_Drops: This is a modification of the Ethernet MAC protocol that does not drop packets. Ethernet is designed to drop packets on excessive re-transmission attempts (16 attempts specifically). While the tree based resolution mechanism in ISS_MAC and ISS_No_Determinism are also designed to drop packets on excessive attempts, the tree mechanism reach such a stage extremely rarely. Since the other MAC protocols being compared, besides Ethernet, do not drop packets, they (including the IDL_MAC) incur more access delays compared to Ethernet. The 'No_Drops' modification was simulated to evaluate the performance of Ethernet MAC as compared to others if there were no packet drops. Ethernet_No_Drops is not a feasible protocol because of the extraordinarily large access delays incurred by a

few packets. It was simulated only to address the author's curiosity. Performance comparisons with Ethernet_No_Drops do not carry much importance.

8.2 Common aspects in simulations

The rest of this chapter, from section 8.3 onwards, describes the various simulations performed for performance evaluation, and the corresponding results. In this section, we briefly discuss some aspects common to all simulation experiments performed.

8.2.1 Load properties

In order to fairly compare protocols, the input loads in the comparison experiments were made to be as identical as possible. The input load in each protocol was matched exactly to the instant of packet arrivals, number of packet arrivals, and packet sizes. The arriving packet sizes in all experiments followed the 'Internet packet size distribution (IPSD)', except where mentioned otherwise. The IPSD is a standard distribution reflecting the packet sizes observed in the Internet. The IPSD was proposed for simulations by the IEEE 802.14 group for their performance evaluation experiments. [44]. The distribution may be found in table III in [23]. The distribution is a discrete one with a mean packet size of 368.1 bytes. The packet sizes and their respective probabilities in IPSD are (64, 128, 256, 512, 1024, 1518 bytes) and (0.6, 0.06, 0.04, 0.02, 0.25, 0.03).

Although the corresponding packets arriving in different networks had identical sizes, the headers were required to be somewhat different for fair comparisons. The base header employed in all protocols was the Ethernet header. However, Ethernet header does not carry any class indicator field. Thus, an additional byte is required in the headers of all protocols that involve the standard 8 priority classes. Actually, just 3 extra bits are required to specify the priority class in the packet headers. In any case, the IDL_MAC simulated in this project involved 1 additional byte in the header besides the Ethernet header. As discussed in chapter 6, CSMA/ISS requires 2 additional bytes in header for protocol operation that includes class specification. Thus, the headers of ISS_MAC, and ISS_No_Determinism involved 2 bytes of additional overhead beyond the Ethernet header overhead. Since the data fields in the MAC protocol data units (PDUs) were identical, the comparison among the protocols is fair.

The load specified in the results compiled below includes the entire MAC PDUs including the preamble and the frame check sequence bytes. The same is true for the utilization values presented below. The entire duration of a valid MAC PDU transmitted is considered for channel utilization computations, and not just the data fields in the packets.

8.2.2 Performance parameters

Following are the parameters discussed in the performance evaluation results in the rest of this chapter.

1. Utilization: This is the throughput normalized with respect to the channel data rate. Thus, it is also the fraction of channel time that is spent transmitting valid data packets. The range is between 0 and 1. In the case of bit errors or other reception errors, *goodput* is defined as the throughput that is successfully received at intended receivers.
2. Access delay: The term 'access delay' for a packet implies the time from when it reaches the head of the queue to the instant its valid transmission begins. Thus defined, it does not include any queuing delays or transmission delays. The time

measured is the time required by the MAC protocol to schedule a packet's transmission once it is ready to be transmitted. The exclusion of queuing delay removes the dependence of the parameter on the arrival process and the existing backlog at the transmitting node. Similarly, the exclusion of transmission delay removes the dependence of the variable on the size of the packet being transmitted. Note that small variations in the mean access delay can significantly impact the net delays experienced by packets. This is so because the access and transmission delays of all the packets backlogged in the queue at the arrival instant are included in the net delays. Also note that with a finite number of nodes, the probability of access delays growing out of bounds is minimal. However, depending on the arrival process, net delays may become infinite despite the access delays being finite.

3. Fairness index: If a fixed amount of a resource is shared among a finite number of entities, the fairness in distribution may be measured in terms of an index called fairness index. It is an index that may have values between 0 and 1, with 1 denoting perfectly fair, and 0 denoting completely unfair. Fairness is expressed in terms of an equitable distribution of resources. In case a fairness rule prescribes more allocation to some entities compared to others in terms of weights, the fairness index is computed by normalizing the allocation with respect to weights. If X_i's represent the normalized allocated resources for individual entities, the fairness index is computed as follows [45].

$$FI = \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n.\left(\sum_{i=1}^{n} X_i^2\right)} \tag{2}$$

Other parameters discussed include the mean, variance, probability mass function (PMF), and probability density function (PDF) for access delays. In order to evaluate the collision avoidance and resolution efficiency, collision rates and PMFs of number of transmission attempts per packet are also evaluated.

For the rest of this chapter, the example implementation of CSMA/ISS that is described in chapter 7, and was modeled for performance evaluation, is referred to as CSMA/ISS.

8.3 A single priority example simulation

Consider the performance of CSMA/ISS in a single priority class environment. In order to bring out the performance gains in CSMA/ISS, consider an example simulation representative of a typical scenario. The simulation set up consists of *16 nodes* that are equally loaded with packet arrivals following an *M/Pareto process*. Network traffic arrivals have been shown to be long range dependent and self-similar in nature [15]. Reference [46] shows that such self-similar traffic may be modeled using an M/Pareto packet arrival process at individual network nodes. M/Pareto traffic involves traffic bursts that arrive with an exponential inter-arrival time, and last for Pareto distributed durations [46]. Within the traffic bursts, packets arrive at constant rates. The parameters of M/Pareto processes can be chosen to model different degrees of burstiness. Burstiness in network traffic is represented by the Hurst parameter [47]. The parameter value for most network traffic of today lies between 0.7 and 0.9. For the simulations in this project, the Hurst parameter value used was 0.75. Returning to the simulation set up description, Figure 14 shows the network input load as it varies with time for 100seconds. The M/Pareto distributions employed were exponential with mean 0.26 seconds, Pareto with delta and gamma as 0.02 and 1.5 seconds respectively, and inter-arrival within bursts as constant 120 microseconds. The priority class of the traffic was chosen to be priority 3, without any loss of generality.

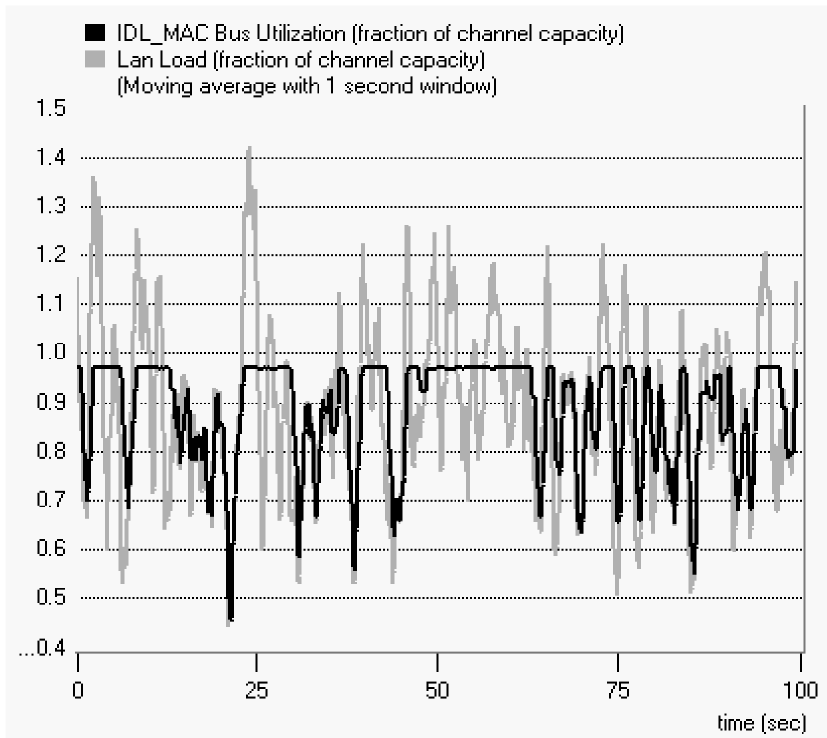


Figure 14. A single priority example simulation: Network Load and IDL_MAC utilization

Figures 15 and 16 show the utilization achieved by the different protocols. We observe that ISS_MAC performs significantly better than Ethernet, and extremely similar to the benchmark ideal scheduling performance. We also observe the there are significant performance gains with state-keeping as ISS_No_Determinism does not perform as well as ISS_MAC.

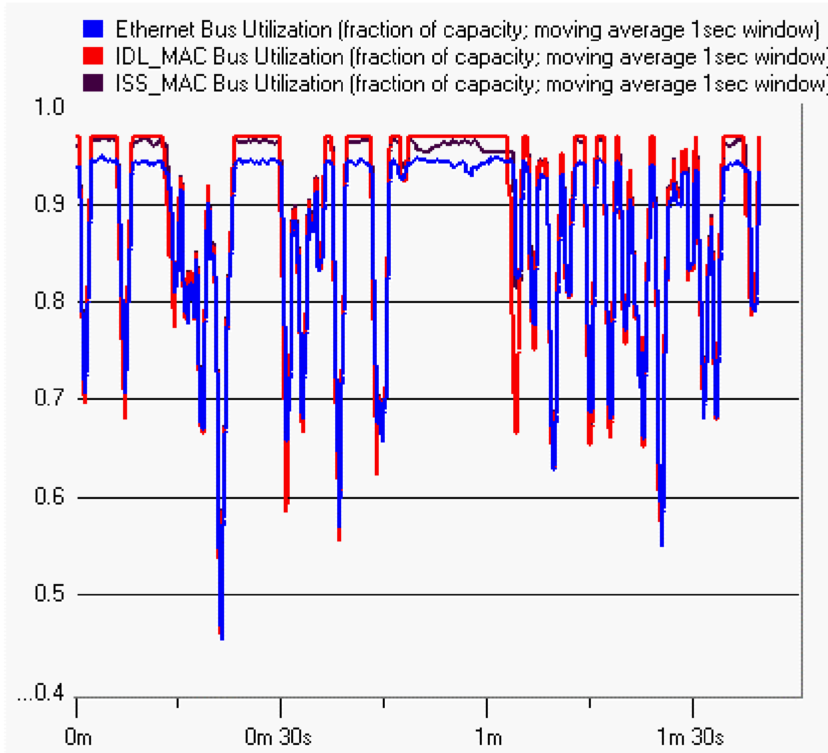


Figure 15. A single priority example simulation: Utilization for different networks

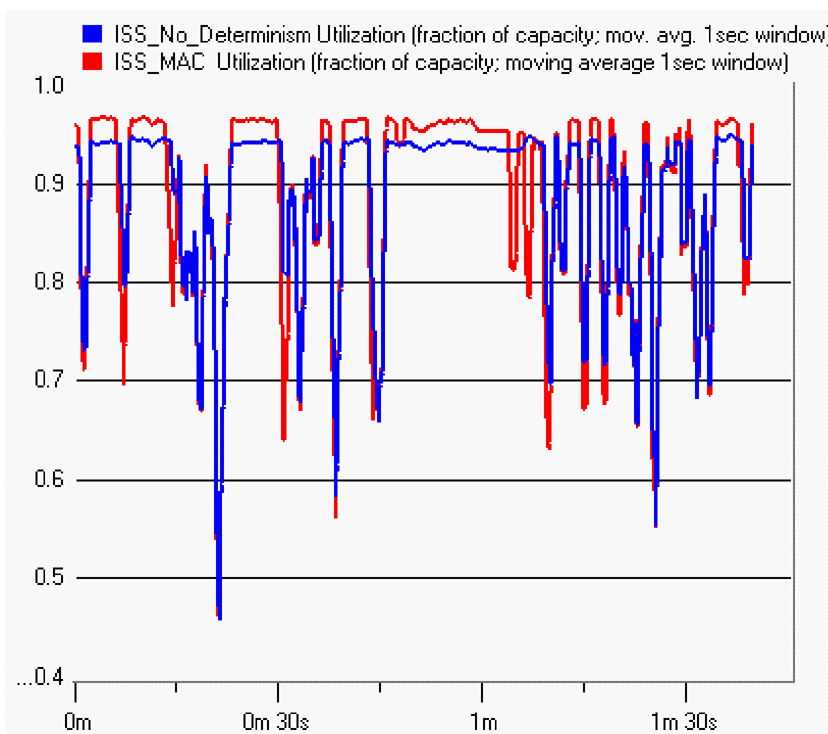


Figure 16. A single priority example simulation: Utilization for ISS_MAC and ISS_No_Determinism

The performance of ISS_MAC in the case of errors is shown in figure 7. We observe that bit errors in packets do not affect the performance of CSMA/ISS significantly. The loss in utilization with a high BER of 1e-5 simply corresponds to the fraction of packets lost due to bit errors. However, the loss in performance due to loss in connectivity is significant. Note that the example implementation described in chapter 7 is not designed for channels with any significant probability of loss in connectivity. However, since the loss in utilization is almost the same in ISS_No_Determinism, and in fact somewhat more than that in ISS_MAC, the cause of performance loss is not the state-keeping process. The performance loss may probably be reduced if the state-keeping process is employed with an access mechanism adapted to connectivity loss environments.

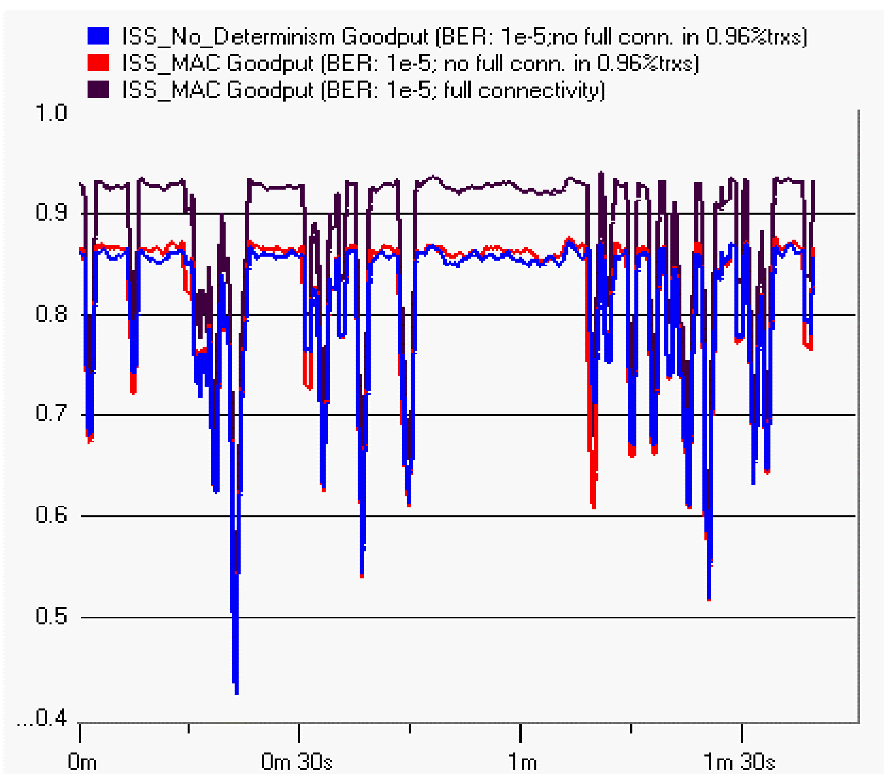


Figure 17. A single priority example simulation: Goodput in error conditions

Access delays

The variation of access delays for the protocols is shown in Figure 18. We observe that while the access delays are the lowest for Ethernet, it drops 0.87% of all packets. For all protocols that do not drop packets ISS_MAC is observed to cause the lowest access delays. Note that the difference in performance among different protocols is prominent at higher loads. As in the case of utilization, significant gains are observed with state-keeping in access delays too.

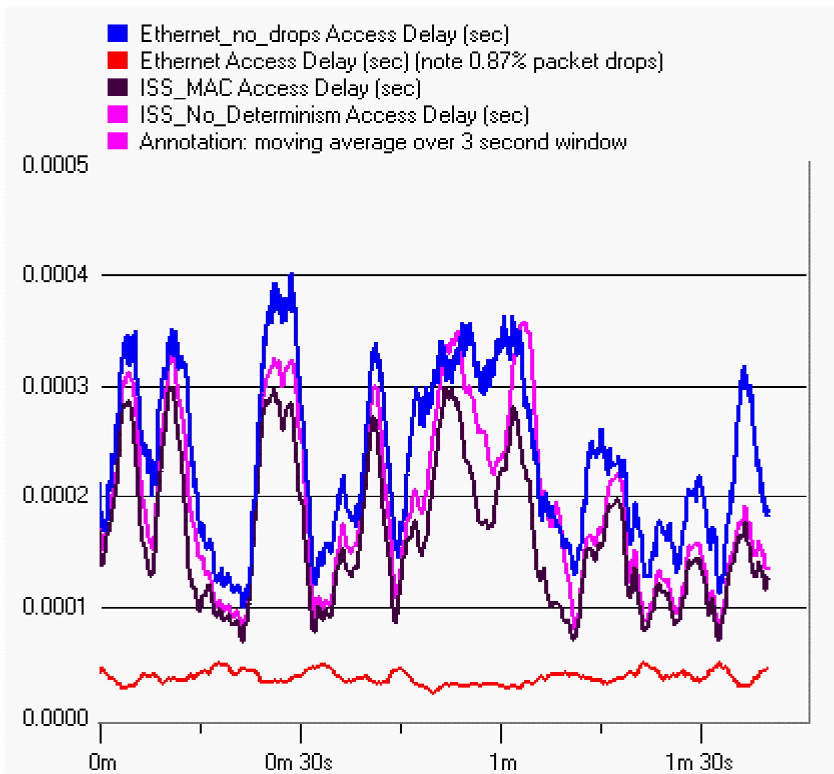
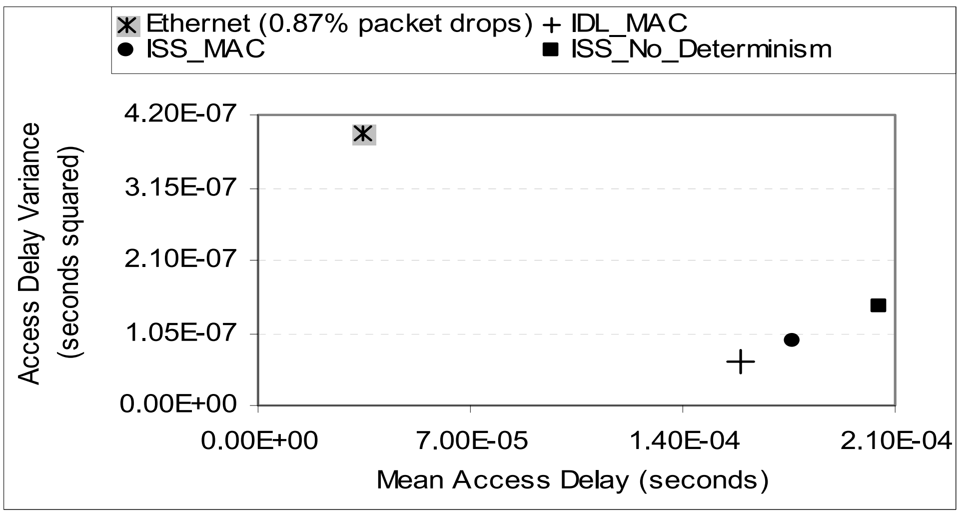


Figure 18. A single priority example simulation: Access delays

The mean and variance of access delays incurred by packets over the course of the entire simulation reflect on the fairness and efficiency performance of protocols. Both the mean and variance should be as low as possible. Figure 19 shows the mean-variance scatter for the example simulation. We observe that the performance of ISS_MAC is the closest to the ideal benchmark performance. We also observe that state-keeping improves both the mean and the variance performance in ISS_MAC. Note that the mean of access delays in Ethernet is smaller than even IDL_MAC. However, the variance is significantly larger, and Ethernet also drops about 0.87% of all packets.



MAC Type	Mean Access Delay	Access Delay Variance
Ethernet_no_drops	2.3580E-04	5.2830E-05
Ethernet	3.4270E-05	3.9450E-07
IDL_MAC	1.5930E-04	6.1700E-08
ISS_MAC	1.7600E-04	9.4000E-08
ISS_No_Determinism	2.0460E-04	1.4440E-07

Figure 19. A single priority example simulation: Mean and variance of access delays

Collision avoidance and resolution performance

Figure 20 and Figure 21 show the collision rates and transmission attempts per packet for the simulation. Note that the collision rate is significantly lower in ISS_MAC than in all other protocols. Prominent state-keeping gains are also observed as ISS_No_Determinism performs significantly worse than ISS_MAC.

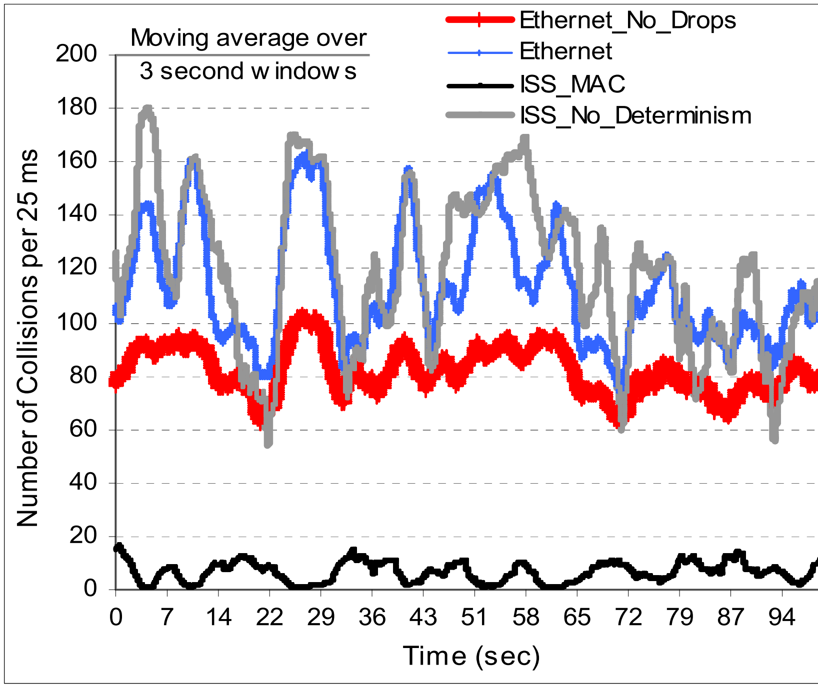
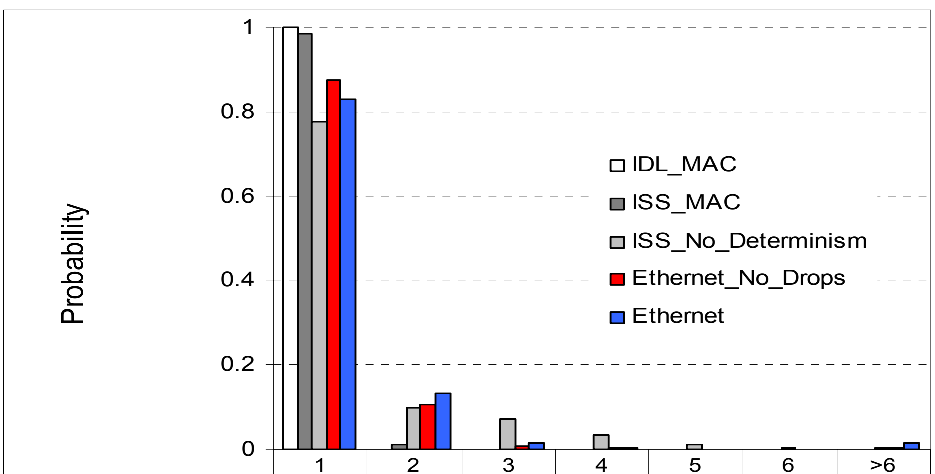


Figure 20. A single priority example simulation: Collision rates



	1	2	3	4	5	6	>6
IDL_MAC	1	0	0	0	0	0	0
ISS_MAC	0.98543	0.01179	0.00187	0.0006	0.0002	6.3E-05	4.2E-05
ISS_No_Determinism	0.77656	0.09985	0.07317	0.03227	0.01198	0.00406	0.00211
Ethernet_No_Drops	0.87688	0.1043	0.00937	0.00343	0.00123	0.00061	0.00418
Ethernet	0.83058	0.13321	0.01493	0.00506	0.00175	0.00095	0.01351

Number of transmission attempts

Figure 21. A single priority example simulation: PMF of number of transmission attempts

ISS_MAC also achieves successful transmissions in the first attempt for significantly large fraction of packets compared to all other protocols (except IDL_MAC). The fraction of packets requiring higher number of attempts is also observed to drops steeply in ISS_MAC. Comparing the transmission attempts required in ISS_MAC versus ISS_No_Determinism, significant state-keeping gains are observed.

8.4 Steady state efficiency

In order to evaluate the steady state performance of ISS_MAC, a number of simulations were undertaken to observe the maximum sustained utilization for different number of nodes in the network. The simulation setup involved all nodes equally loaded with an M/Pareto packet arrival process with a Hurst parameter of 0.75. As the load was increased, the maximum sustained utilization achieved by the protocol was observed. The utilization was observed averaged over 10 second intervals, and the mean of 5 simulations was noted as the value. The maximum utilizations for particular scenarios were observed to be extremely similar across simulation runs. Figure 22 shows the results. While the protocols that do not involve state-keeping are observed to cause lower maximum utilizations as the number of nodes increases, ISS_MAC is observed to maintain its performance. The efficiency of contention-based access is known to reduce as the number of contending nodes increase [9][21]. ISS_MAC, through its state-keeping, allows only small subsets of nodes to contend together. Also, the contending subsets are estimated to have only a single node backlogged. Thus, regardless of the number of backlogged nodes in the network, ISS_MAC maintains its high throughput. The utilization performance of ISS_MAC is also observed to extremely similar to the IDL_MAC performance.

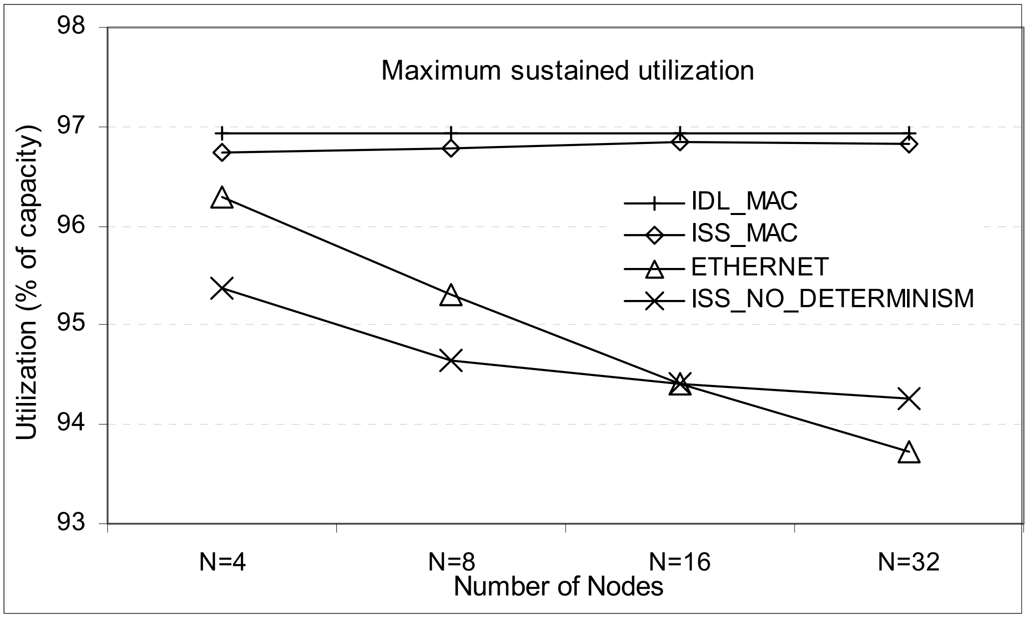


Figure 22. Maximum utilization variation with number of nodes

Note that the minor improvement in maximum utilization with increasing nodes is ISS_MAC may be attributed to the decrease in the ring traversal overhead per transmission, as number of nodes increases.

Access delays

Figure 23 through Figure 26 show the mean variance scatter for one of the simulations that achieved the maximum utilizations for each case described in Figure 22. The mean network load for each of the simulations was marginally greater than the channel capacity. We observe that regardless of the number of nodes in the network, the mean and variance of access delays achieved by ISS_MAC are the closest to those achieved by the IDL_MAC. The ISS_MAC performance is extremely similar to the IDL_MAC performance, and significantly better than the performance of Ethernet and ISS_No_Determinism.

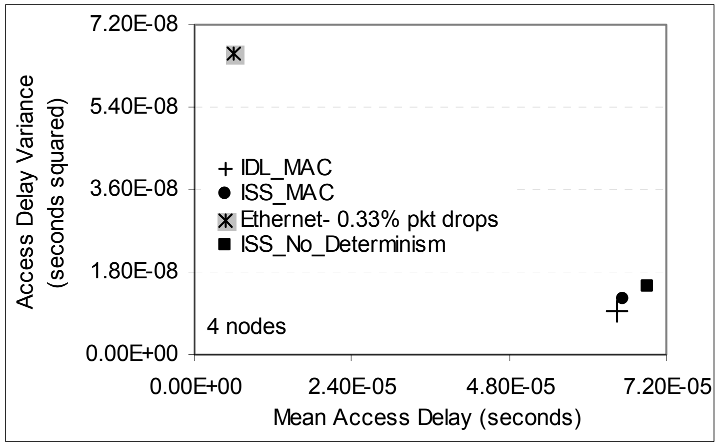


Figure 23. Mean variance scatter of access delays; maximum utilization scenario; 4 nodes

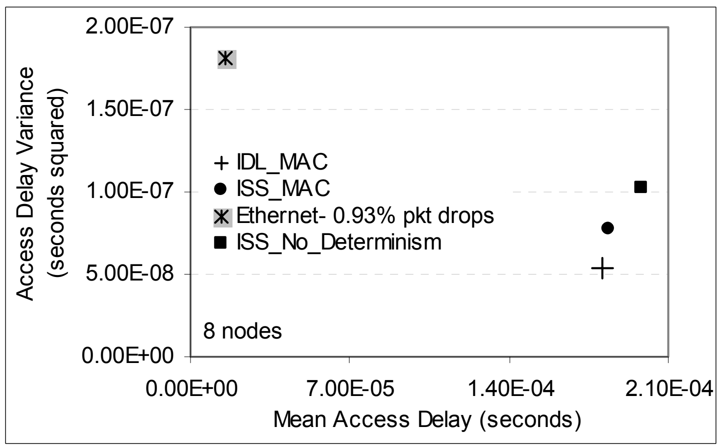


Figure 24. Mean variance scatter of access delays; maximum utilization scenario; 8 nodes

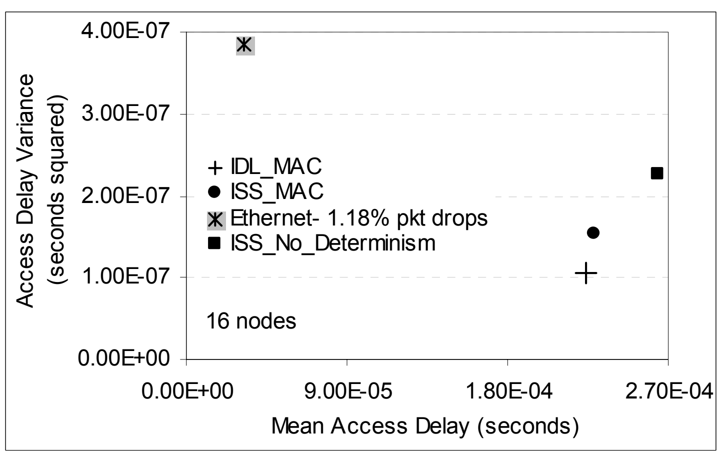


Figure 25. Mean variance scatter of access delays; maximum utilization scenario; 16 nodes

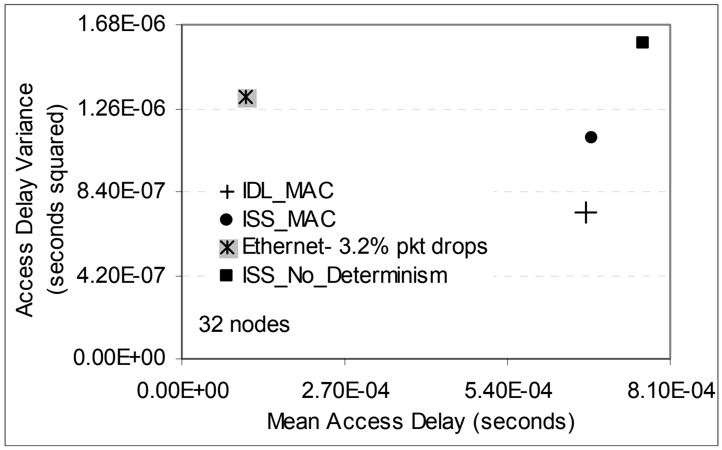


Figure 26. Mean variance scatter of access delays; typical maximum utilization scenario; 32 nodes

8.5 A Multi-priority example simulation

In this section, we present the results from an example simulation scenario with traffic in multiple priority classes. The simulation scenario is designed to bring out the capability of CSMA/ISS for channel efficient hierarchical fairness. It is as follows.

- All traffic classes in all nodes are equally loaded with exponential on-off arrivals when active.
- There are 16 nodes in the network. Traffic classes at nodes become active and inactive at various times, as described below.
- The load in each active class at each node is such that when all 16 nodes are active, a class has a mean net load that is marginally greater than 0.3 (normalized with respect to channel capacity of 100 Mbps). The value of 0.3 is the capacity available for the highest priority (i.e. priority 7) according to the weights chosen for weighted fair queuing in ISS_MAC. Note that if less than 16 nodes are active, the load proportionally reduces. The reason for the choice of load greater than 0.3 in all

priorities is to evaluate the weighted fairness when all classes are equally loaded. The hierarchical fairness capabilities of ISS_MAC are evaluated by varying the number of nodes that are active, as described below. The variation of load in each class with time is shown in Figure 27.

- At time 0, all nodes become active in all traffic classes except class 0 (lowest), 4 (middle), and 7 (highest). Only 8 out of 16 nodes become active in classes 0,4, and 7.
- The remaining 8 nodes that are inactive at time 0 in classes 7,4, and 0 become active at times 10, 15, and 20 seconds respectively.
- All 16 nodes in PRI 7 to 1 become inactive at times between 25 seconds and 55 seconds, with one class becoming inactive every 5 seconds, as shown in Figure 27. The order in which the classes become inactive is 7 to 1 in descending order.

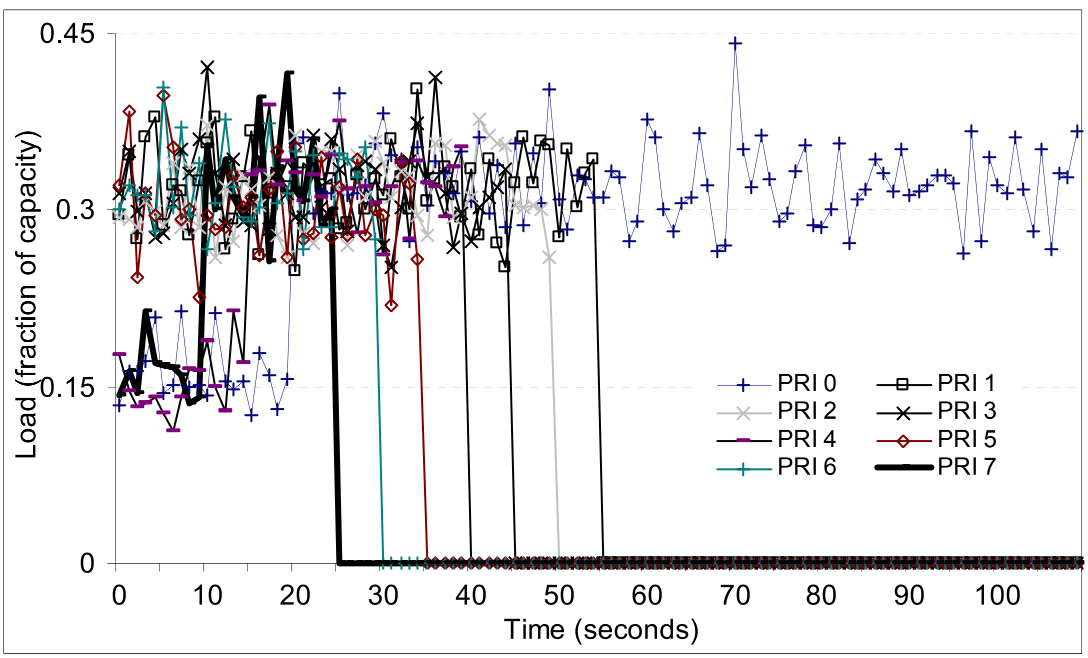


Figure 27. A multi-priority example simulation: Load in various priorities

In order to evaluate if the utilization achieved by various classes in the network is according to fairness rules, consider the weights of each class again. The weights for classes 7 to 0, as defined in chapter 6, are 15,11,8,6,4,3,2, and 1 respectively. By the weighted fairness rules, if all classes are sufficiently loaded, *classes 7 to 0 should achieve utilizations of 0.3, 0.22, 0.16, 0.12, 0.08, 0.06, 0.04, and 0.02 respectively*. These are computed as ratios of weights of each class with the total weights of all classes. The fairness rules also imply that if higher priorities are not loaded enough to use their available capacity, the lower priorities share the extra capacity in proportion of their weights [5].

Figure 28 and Figure 29 show the utilizations achieved by various classes, and the network as a whole in the simulation. Figure 28 shows the variation of loads and utilizations in the first 25 seconds for priorities 0,4, and 7. We observe the following from the figures.

- From time 0 seconds to 10, the load in class 7 is less than the available capacity of 0.3, thus load and utilization are identical for the class. The rest of the capacity is distributed among classes 6 to 0 according to their weights. The loads in classes 6 to 0 are more than their available capacity. The loads in classes 0 and 4 are about half the loads in other classes, yet the utilizations achieved are proportional to the weights and do not depend on the load values.
- At time equal to 10 seconds, the load in class 7 also increases beyond its available capacity. The class achieves its available capacity of about 0.3, and the rest of classes lose their extra access. From 10 seconds to 25 seconds, the loads in all classes are

more than their available capacities. The utilizations achieved by the classes are observed to be proportional to their weights.

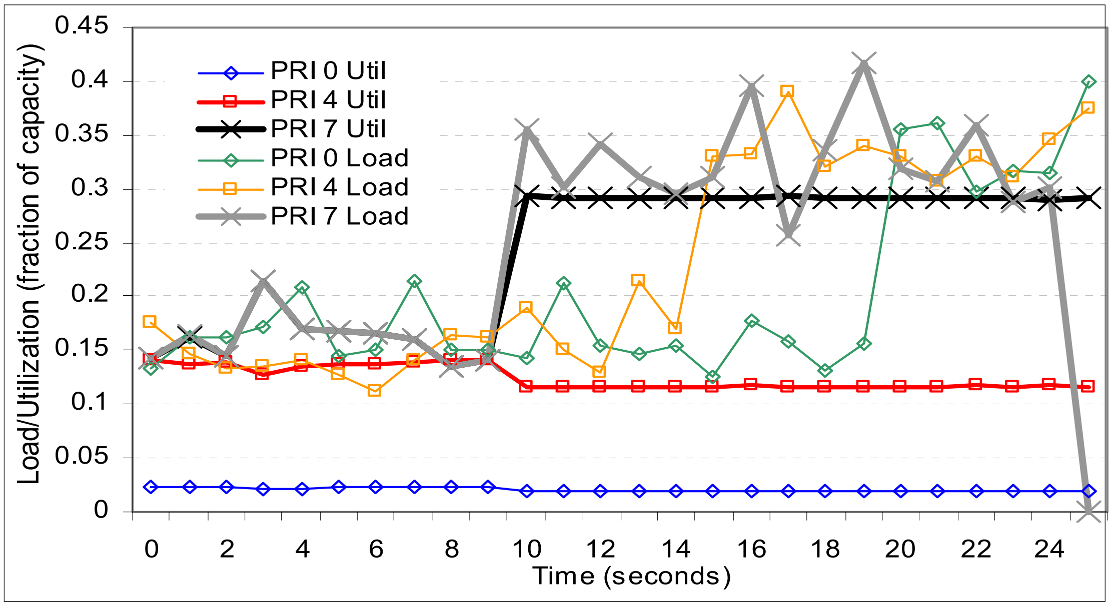


Figure 28. Multi-priority example simulation: Load and Utilization in selected classes in the first 25 seconds

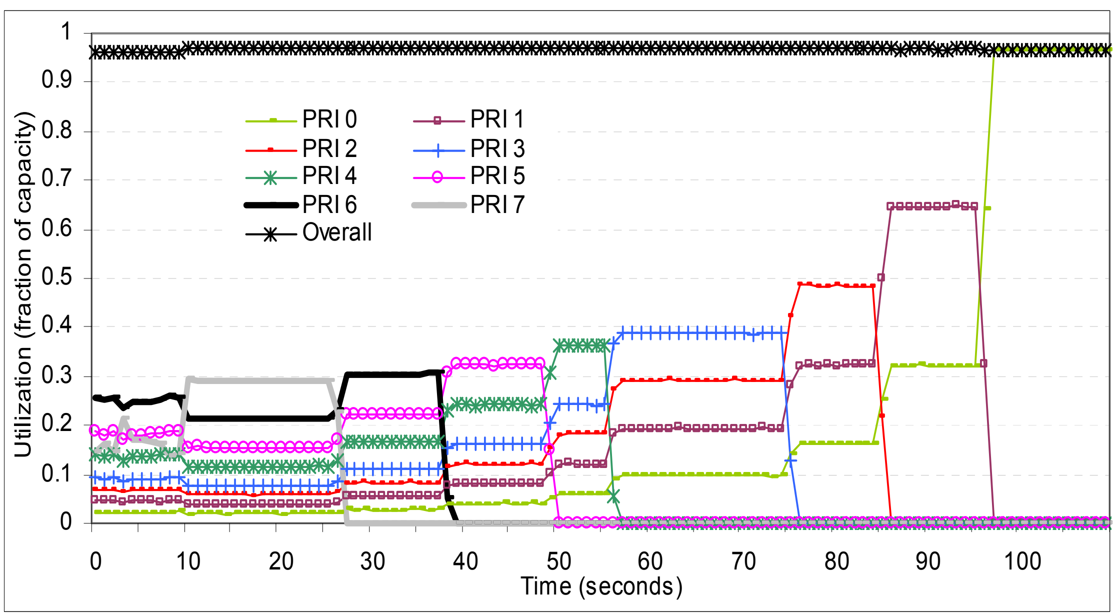


Figure 29. Multi-priority example simulation: Utilization

- Hierarchical fairness is observed as the number of nodes active in classes 4 and 0 double from 8 to 16 at times 15 seconds and 20 seconds respectively. The doubling of loads, and the initiation of new flows in these classes does not affect the net utilizations achieved by any of the classes. All classes are loaded more than their capacities already, and the additional load does not affect the utilizations achieved by the classes. The access achieved by nodes within classes 0 and 4 do get affected as the number of active nodes double. This may be observed in Figure 30 and Figure 31. At 15 and 20 seconds simulation time, the mean access delays incurred by nodes in classes 4 and 0 respectively are observed to double. This is observed in reaction to the doubling of the number of nodes active in the classes. However, the access delays incurred by nodes in other classes are unaffected at these instants. This disconnection between the fairness within and across classes establishes hierarchical fairness.
- As classes begin to become inactive from the simulation time of 25 seconds onwards, the remaining active priorities are observed to use the available capacity in proportion to their weights. This again establishes the correct operation of weighted fair scheduling in ISS_MAC. Note that the instants at which the utilizations become 0 for classes are later than the instants at which their loads become 0. This is because of the backlog of traffic being accumulated, as the available capacities for classes are lower than the loads when all classes are active in the beginning.
- Despite varying loads that are significantly higher than the available capacities, classes utilize the channel according to their weights, and with a high overall utilization. The overall utilization is observed to be about 96%, and almost the same as the maximum utilization achieved by ISS_MAC in a single priority class scenario

(see section 8.4). Thus, fairness is achieved with efficient channel utilization in ISS_MAC.

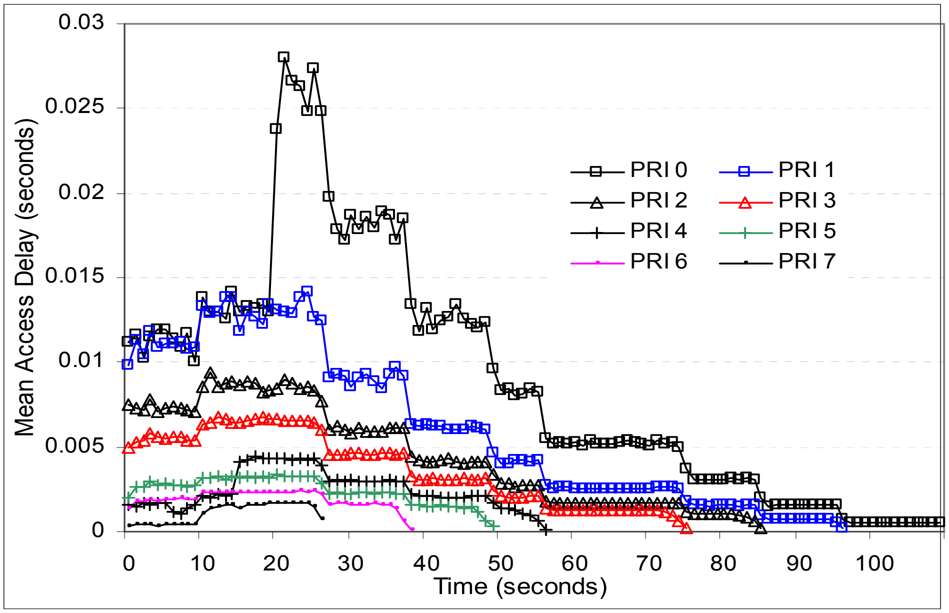


Figure 30. Multi-priority example simulation: Access delays

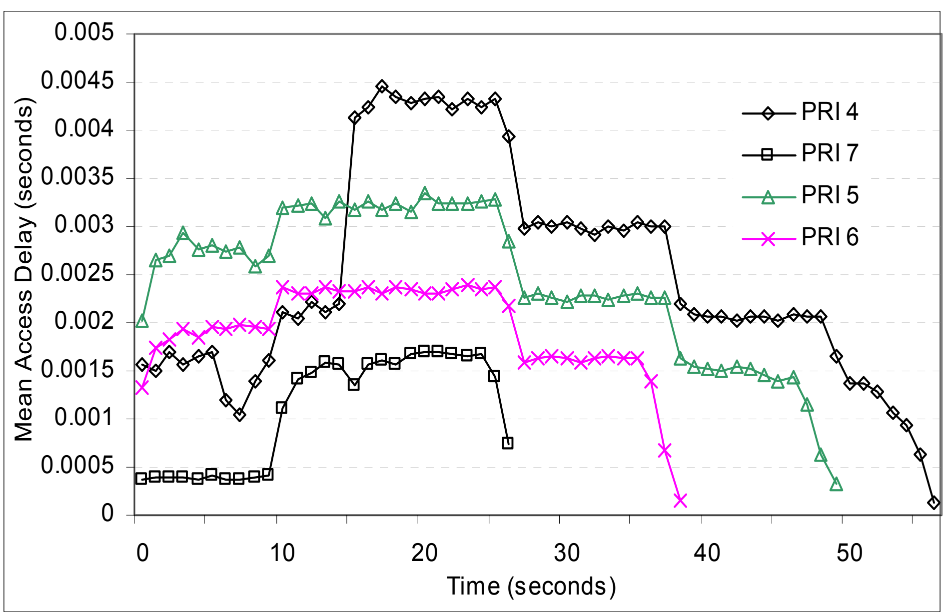


Figure 31. Multi-priority example simulation: Access delays for selected classes

In the simulation, the mean access delays incurred by nodes in various classes reflect the loads in classes, and the fair sharing rules. We observe that, as expected, the mean access delays vary according to the utilization achieved by the respective classes, and the number of active nodes in the classes (Figure 30 and Figure 31).

8.6 Hierarchical fairness over small time scales

In this section, we present the results of fairness evaluation experiments for ISS_MAC. Fairness over small scales of time implies fairness over large scales of time. Thus, the results in this section hold for all scales of time larger than the ones presented here. The salient features of the simulations performed are as follows.

- Two identical simulations were performed, one for 30 seconds, and the other for 300 seconds. The order of magnitude difference in the times is due to the order of magnitude difference in the scales of time over which fairness is evaluated.
- Each simulation period is divided into 100 equal intervals. The intervals are 0.3 second and 3 seconds respectively for the two simulations. Fairness in channel utilization is evaluated for each such interval through the fairness index (FI) of allocation. The allocated resource for the FI computation is the amount of transmitted data in the intervals by individual entities (classes or nodes).
- The simulation scenarios involve 16 nodes, with all classes in all nodes equally loaded with packet arrivals every 100 microseconds. As in all simulations, the packet sizes follow the 'Internet Packet Size Distribution (IPSD)' described in 8.2.1.

Figure 32 and Figure 33 show the variation of fairness indices for channel sharing across the priority classes. We observe that strong fairness is achieved with the FI values extremely close to 1, for both 0.3 second and 3 second intervals. As expected, the fairness achieved over longer time scale is stronger than that achieved over the smaller interval.

Figure 34 through Figure 41 show the fairness indices for capacity sharing by nodes within specific classes. For classes 7 to 4, the indices are shown for sharing over 0.3 second intervals. For classes 3 to 0, they are shown for sharing over 3 second intervals. The main reason for the difference is that the weights of the lowest and the highest priority classes are an order of magnitude different (15 and 1). This causes more packet transmission in higher priorities is the same interval of time. Thus, the indices for higher priorities are shown for the smaller interval of 0.3 seconds. In Figure 34 to Figure 41, we observe that strong fairness is achieved in channel sharing among different nodes within each class. The fairness indices values observed are extremely close to 1. Stronger fairness is observed as the indices are computed over larger number of packet transmissions (caused by larger time interval and/or higher available capacity according to weights). Since lower priorities have less data transmitted compared to higher priorities for identical time intervals, the fairness indices are observed to be lower.

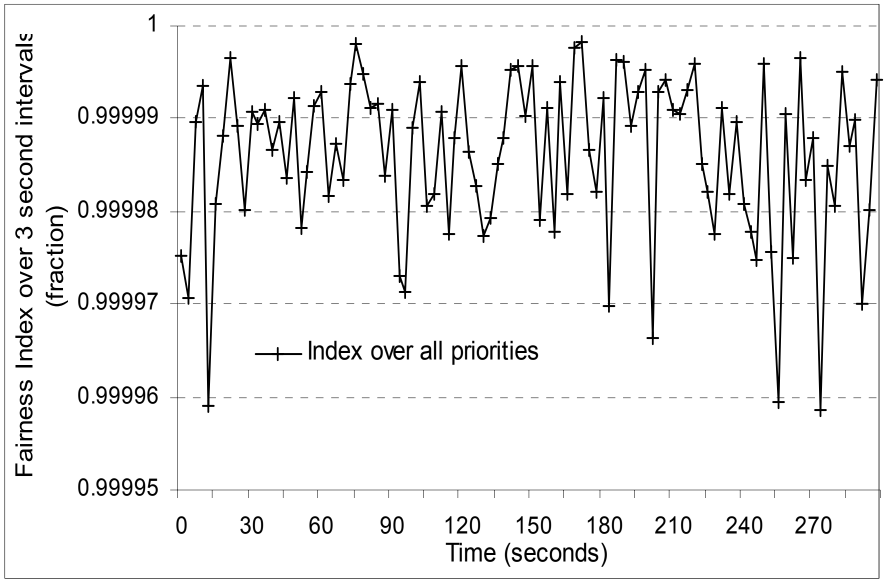


Figure 32. Fairness evaluation: fairness index for sharing across classes; 3 second intervals

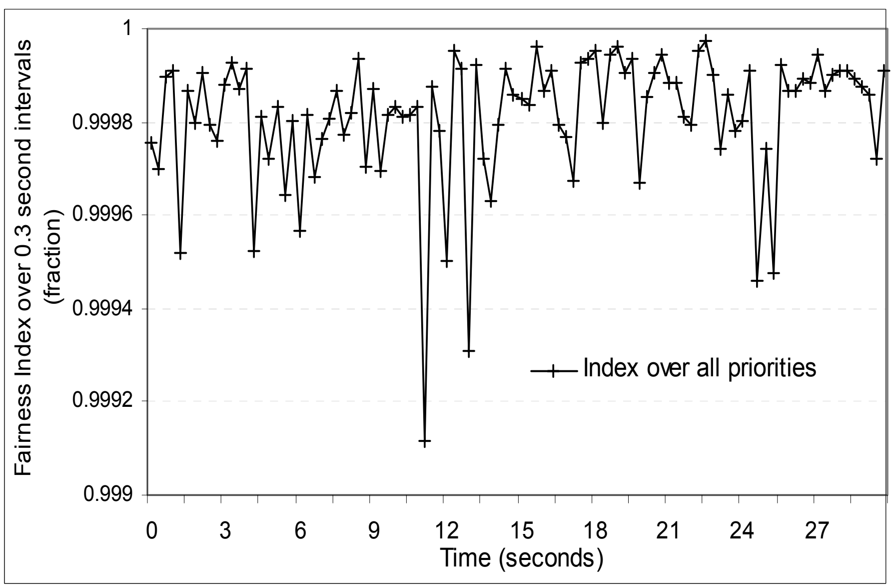


Figure 33. Fairness evaluation: fairness index for sharing across classes; 0.3 second intervals

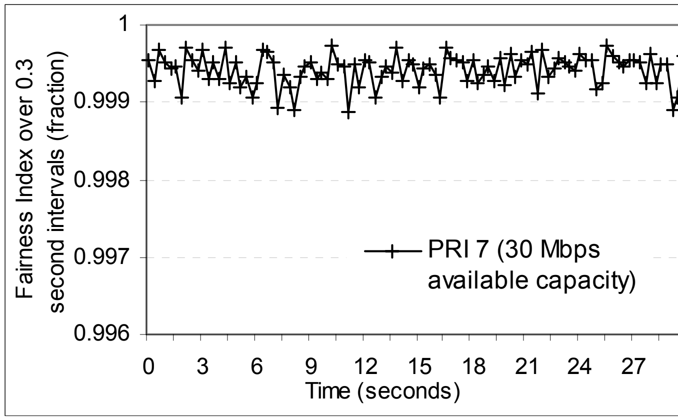


Figure 34. Fairness evaluation: fairness index for sharing within class 7; 0.3 second intervals

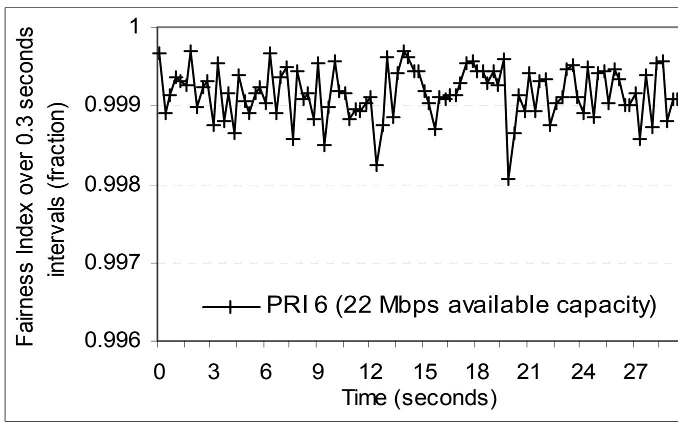


Figure 35. Fairness evaluation: fairness index for sharing within class 6; 0.3 second intervals

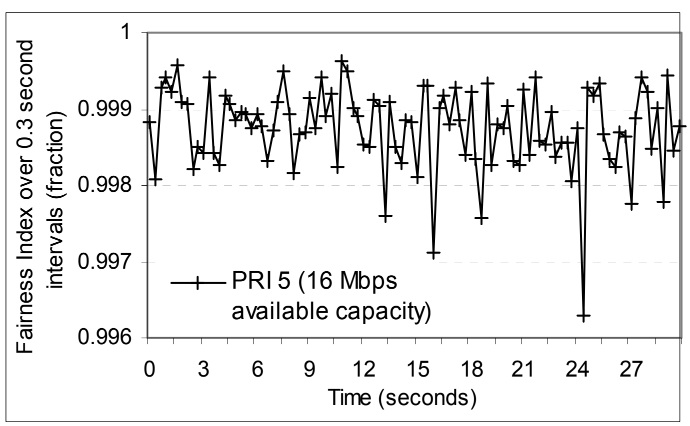


Figure 36. Fairness evaluation: fairness index for sharing within class 5; 0.3 second intervals

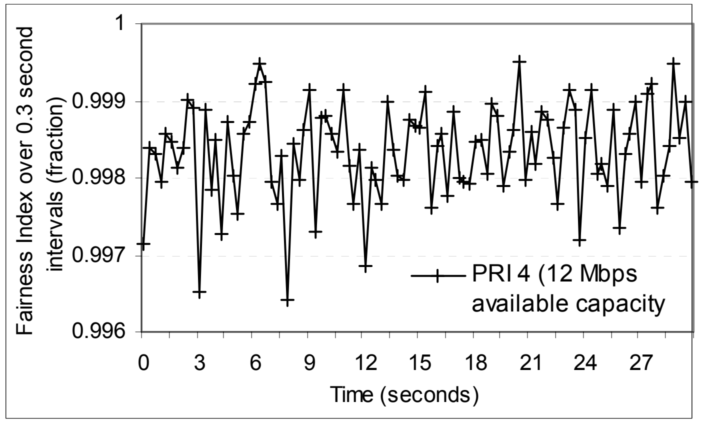


Figure 37. Fairness evaluation: fairness index for sharing within class 4; 0.3 second intervals

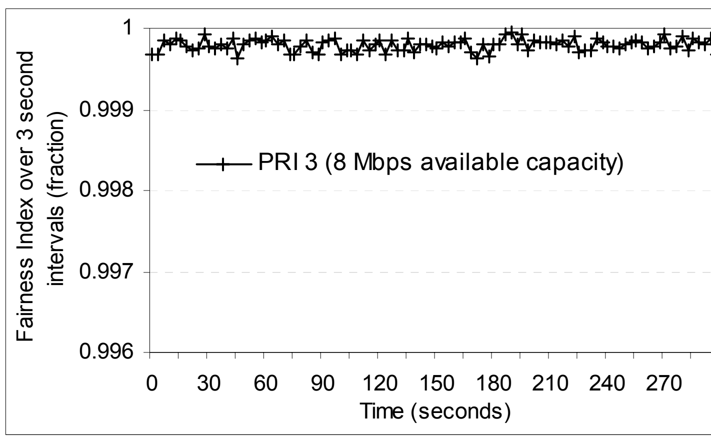


Figure 38. Fairness evaluation: fairness index for sharing within class 3; 3 second intervals

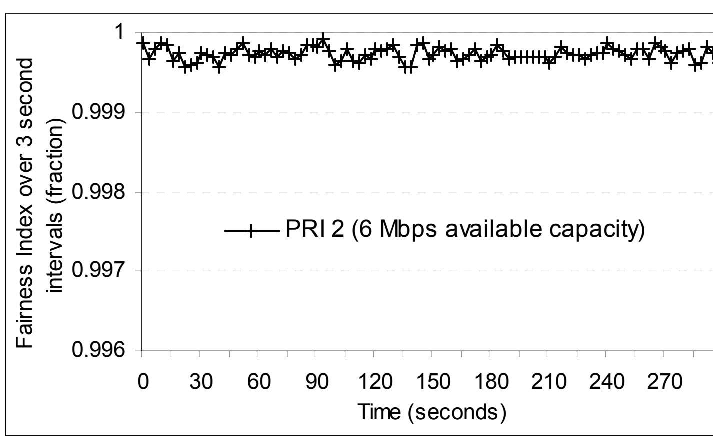


Figure 39. Fairness evaluation: fairness index for sharing within class 2; 3 second intervals

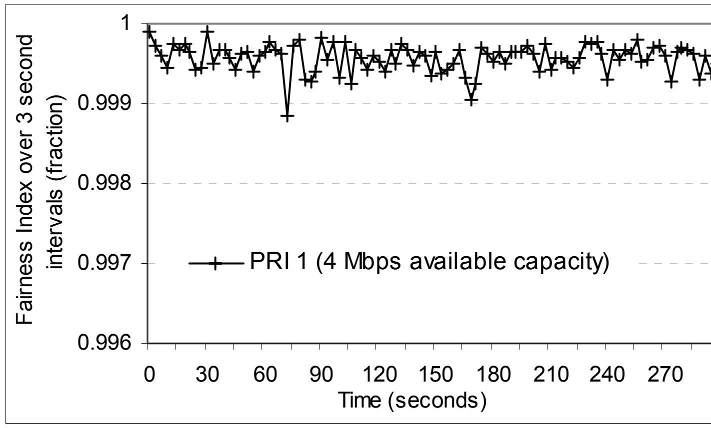


Figure 40. Fairness evaluation: fairness index for sharing within class 1; 3 second intervals

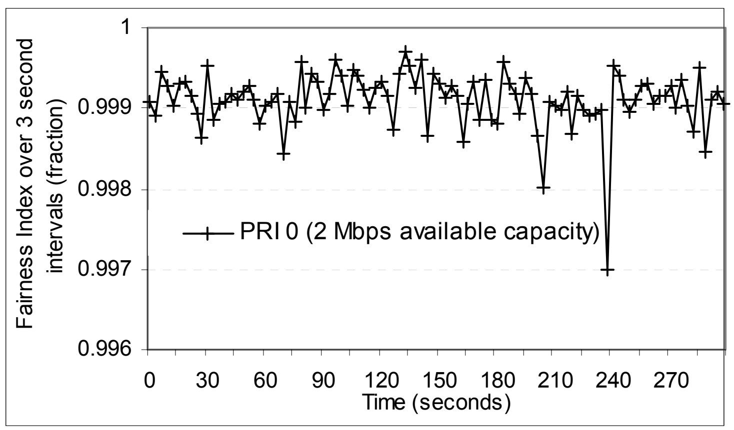


Figure 41. Fairness evaluation: fairness index for sharing within class 0; 3 second intervals

8.7 A typical scenario example

All the simulation experiments discussed above have been designed to investigate specific aspects of CSMA/ISS's performance. In this section, we present a simulation scenario representing a typical load environment for QoS enabled MAC protocols. Load in different priorities may be different, and despite lower weights, load in lower priorities may be higher than in higher priorities. Also, the packet arrival process is different for different traffic types. The following scenario was simulated to investigate performance in such cases.

- All classes except class 2 and 7 have traffic. There are 16 nodes in the network, and each node has traffic in multiple classes.
- Traffic in priority class 6 consists of 16 nodes engaged in low load voice traffic. The arrival process involves packets uniformly sized between 90 to 120 bytes with an inter-arrival time that is uniformly distributed between 19 to 21 milliseconds. The arrival process leads to approximately constant rate traffic.
- Traffic in class 5 consists of 8 nodes involved in low bit rate video conferencing traffic. The arrival process involves constant sized 348 byte packets with constant inter-arrival time of 1 millisecond during the burst *on* periods. The process involves bursts with inter-arrival times uniformly distributed between 99 and 101 milliseconds and batch duration that are uniformly distributed between 11.9 and 12.1 milliseconds. The arrival process leads to approximately constant rate traffic when considered at time scales of the order of seconds.
- Traffic in class 4 consists of 8 nodes involved in moderate quality streaming media traffic. The arrival process is modeled as M/Pareto with burst inter-arrivals with a mean of 10 seconds and burst durations as Pareto (2,1.5). The packet sizes are modeled as uniformly distributed between 500 and 1480 bytes, and the inter-arrival times in the burst *on* periods are constant 0.01 seconds.
- Classes 3, 1, and 0 consist of M/Pareto modeled bursty traffic. Classes 3 and 1 are modeled with 8 nodes active, and class 0 with 16 nodes active. The M/Pareto parameters chosen were mean burst inter-arrival time of 1 second, delta and gamma

of Pareto process as 0.2 and 1.5 respectively, the packet inter-arrivals at constant 1.2 millisecond intervals, and packet size distribution of IPSD (see 8.2.1).

Figure 42 and Figure 43 show the loads in the classes, and the overall load in the network. The net network load is observed to be somewhat greater than 1. The heaviest load is observed in class 0. The loads in other classes are observed to be progressively smaller with increasing priorities. Figure 44 and Figure 45 show the corresponding utilizations achieved. The utilization values for all classes except class 0 are observed to follow the corresponding load values. The loss in utilization for the lowest priority class is in agreement with the fairness rules. The net utilization is observed to be high and greater than 90%. With the net load in the network greater than the capacity for majority of the time, the net utilization is not observed to be affected by the varying loads in different classes. The observed utilization performance is as expected according to the weighted fairness rules.

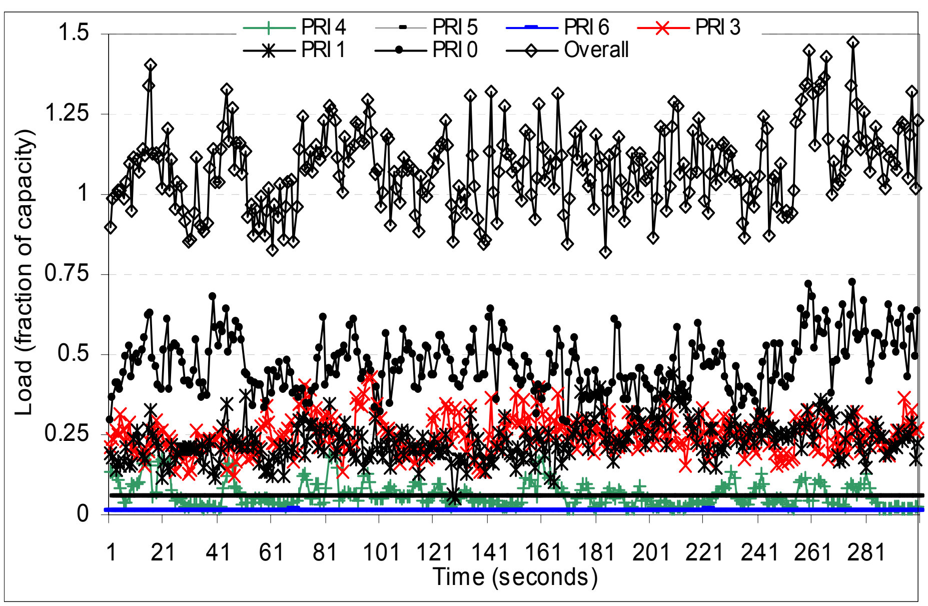


Figure 42. Typical traffic scenario: Load

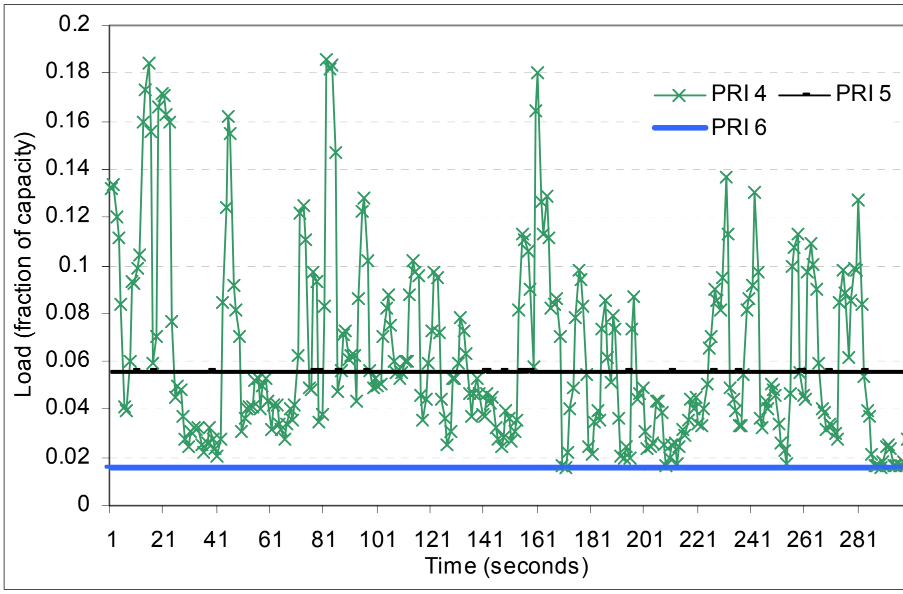


Figure 43. Typical traffic scenario: Load in selected priorities

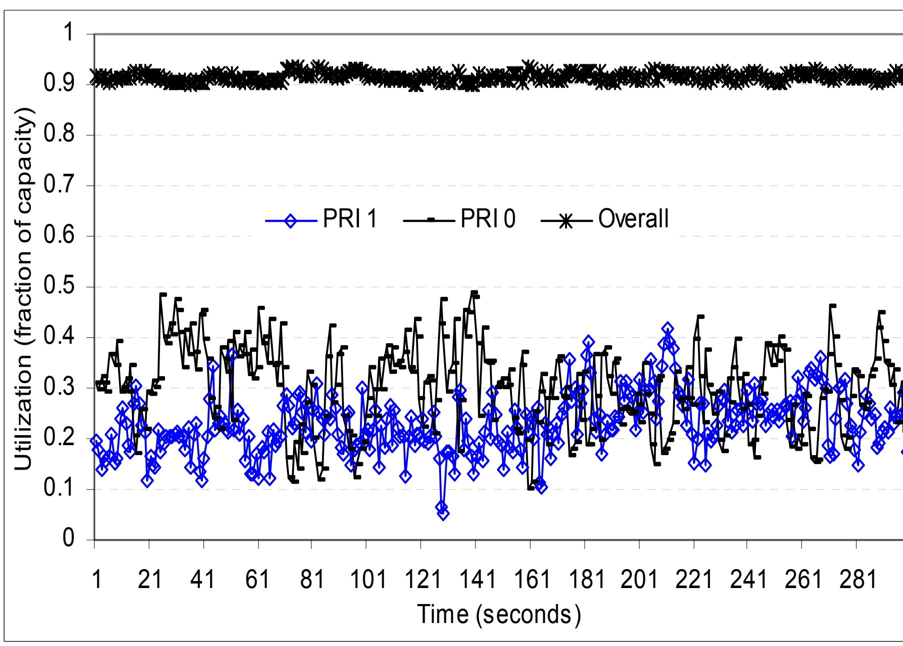


Figure 44. Typical traffic scenario: Utilization in the network and in selected classes

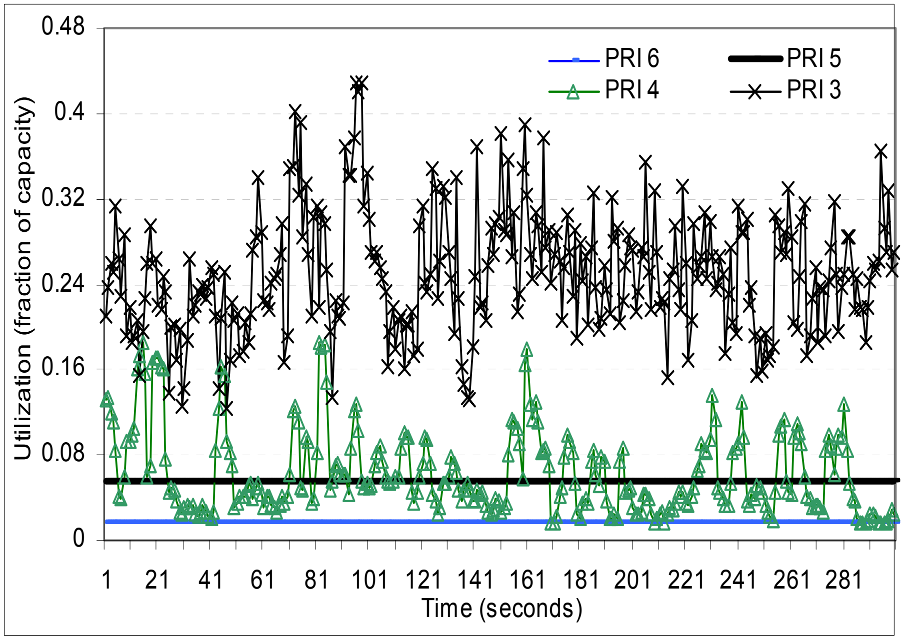


Figure 45. Typical traffic scenario: Utilization in selected classes

Access delays

In order to further verify the operation of ISS_MAC in the given scenario, consider the PDFs of the access delays as presented in Figure 46 and Figure 47. We observe that the PDFs for lower priority classes may have higher densities over smaller access delay values than higher priorities depending on the traffic load and the number of nodes involved in each. If the number of nodes active in a class is small, the access delays incurred can be small despite low available channel capacities. If a large amount of traffic is served in a lower priority as compared to a higher priority, a significant fraction of lower priority packets are not delayed due to higher priority packets. This is so because during the transmission of such a fraction of packets, there may be no higher priority backlog. During these times, if the traffic in other classes reduces or the number of nodes involved reduces, the access delays are lower. The observed access delay performance is as expected with the weighted fairness rules.

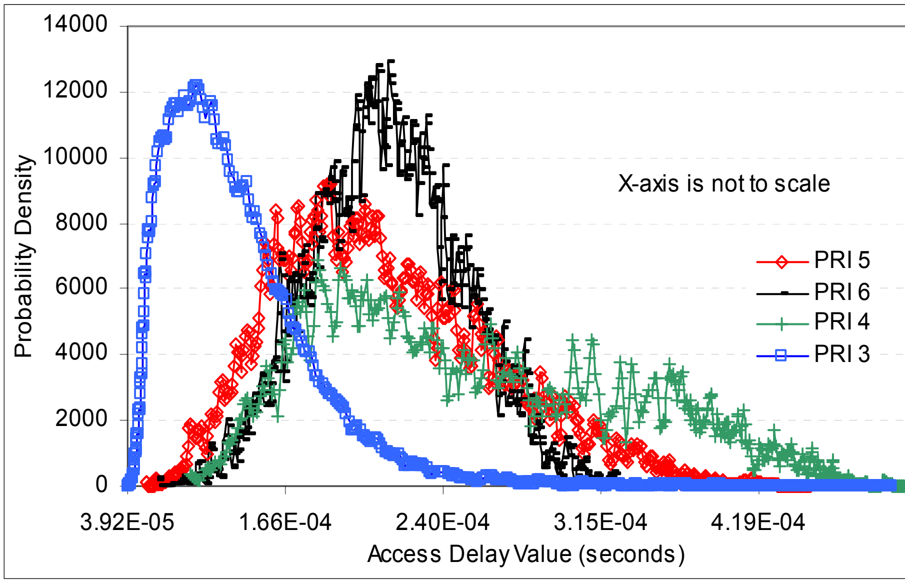


Figure 46. Typical traffic scenario: PDFs of access delays for classes 3 to 6

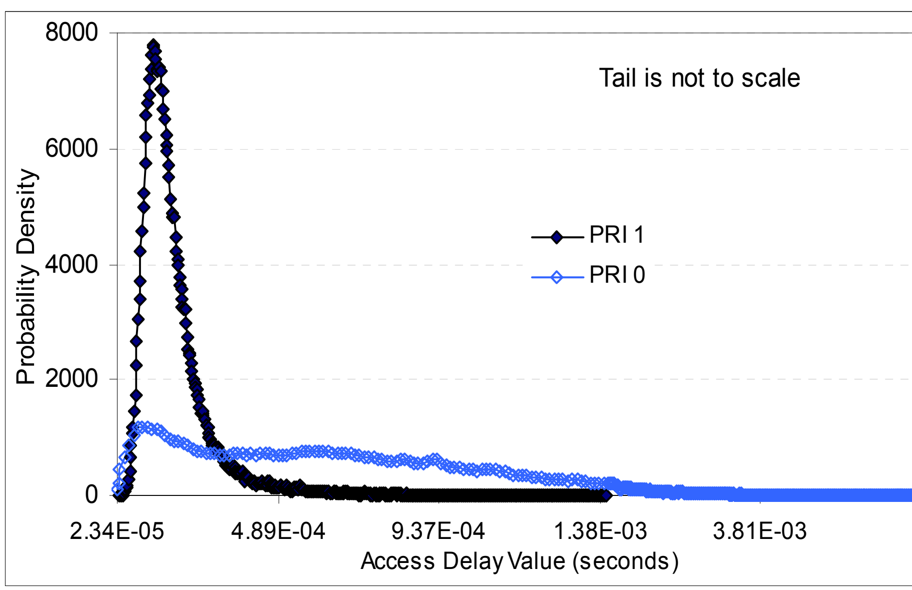


Figure 47. Typical traffic scenario: PDF of access delays for classes 0 and 1

8.8 Performance evaluation summary

In this chapter we discussed the results of the performance evaluation experiments undertaken for the example implementation of CSMA/ISS described in chapter 7. The protocol, referred to as ISS_MAC, was observed to achieve the design objectives of CSMA/ISS. The protocol was observed to achieve strong hierarchical fairness with high channel utilization efficiency. The performance was observed to be extremely close to the ideal scheduling benchmark. ISS_MAC was observed to be more efficient and fair than Ethernet. Substantial performance gains were observed due to state-keeping in ISS_MAC. ISS_MAC outperformed ISS_No_Determinism, a protocol identical to ISS_MAC but without state-keeping. Since ISS_No_Determinism is a simple MAC protocol with ternary tree based collision resolution, ISS_MAC was observed to perform better compared to classical tree based contention resolution protocols. The queue and arrival rate tracking mechanisms in CSMA/ISS were also observed to effectively avoid collisions, and resolve the ones that happen in an expedited manner. The performance results show that the state-keeping approach in CSMA/ISS is a useful approach to achieve better fairness and efficiency in CSMA networks at the cost of moderate processing and storage overhead.

CHAPTER 9

CONCLUSION AND FUTURE DIRECTIONS

In the previous chapters, we presented how fairness-efficiency tradeoff is the fundamental research problem for QoS MAC design engineers, and proposed a framework to improve performance in CSMA networks. We presented the definition, objectives, and architectures of QoS implementations in MAC protocols. We discussed how an inherent fairness-efficiency tradeoff is present in all current QoS MAC architectures. There have not been any comprehensive QoS MAC architectures that employed state-keeping on explicit and implicit MAC feedback to improve both fairness and efficiency. CSMA with implicit scheduling through state-keeping (CSMA/ISS) framework was proposed for carrier sensing networks to address the problem. CSMA/ISS is a distributed control framework based on tracking of queues and arrival rates at all active nodes. While the framework is proposed as distributed control, state-keeping gains may be achieved in centrally controlled MAC environments as well. The tracking and estimation are performed to save channel time in information exchange or that spent in distributed fairness mechanisms. The idea is to begin with default states and update them based on feedback, which may be either implicit through channel activity or explicit in various packets. Fairness is proposed to be implemented in a way similar to processor sharing. Within each traffic class, while the state-keeping builds up ordered list of active nodes, fairness is proposed by allowing equal access to these active nodes in order. Fairness across priorities is proposed to be weight based. It is proposed to be implemented by allowing equal access to the independent MAC state of each priority, a number of times equal to their weights. The proposed framework may be employed along with classical QoS MAC approaches, and needs to be adapted to different CSMA channels according to their reliability and connectivity properties. As the reliability and connectivity in channel suffer, the performance gains from implicit feedback reduce. In such cases more explicit feedback may be employed compared to implicit. The framework was shown to involve a moderate storage and processing overhead. An example MAC protocol for a reliable, fully connected wired network was modeled and simulated to evaluate the performance improvements achievable from CSMA/ISS. Significant performance improvements were observed due to state-keeping. CSMA/ISS was observed to approach Ideal Scheduling performance, both in terms of fairness and efficiency. Simultaneous improvements in fairness and efficiency were shown possible at the cost of moderate storage and processing overhead.

The work on CSMA/ISS presented here suggests a number of future directions of interest. While our experiments prove the proposed concept, a wireless network adaptation of CSMA/ISS, and the quantification of performance improvements therein are the immediate future directions of interest. Some suggestions for a high performance wireless implementation were presented in previous chapters. The quantification of performance improvements with CSMA/ISS as compared to other distributed fairness schemes such distributed fair scheduling (DFS) and Blackburst is also a potential area for immediate future research. Within the proposed framework itself, the structure of the increase-decrease function along which queues and traffic arrival rates are tracked is open to research.

This work presents a novel approach to MAC QoS in CSMA networks, with QoS performance improvements at the cost of moderate storage and processing overhead.

REFERENCES

[1] G. Armitage, *Quality of Service in IP Networks: Foundations for a Multi-Service Internet*, 1st edition, Indianapolis IN, Macmillan Technical Publishers, 2000.

[2] R. Rom and M. Sidi, *Multiple Access Protocols,* New York: Springer Verlag, 1990.

[3] E. H. Frank and J. Holloway, "Connecting the Home With a Phone Line Network Chip Set," *IEEE Micro*, pp.27–38, March-April 2000.

[4] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e wireless LAN for QoS," *Proceedings of the European Wireless*, February 2002.

[5] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, pp. 344-357, June 1993.

[6] A. Demers, S. Keshav, and S. Shenker, " Analysis and Simulation of a Fair Queuing Algorithm," *In Proc. of Symposium on Communications Architectures and Protocols, ACM SIGCOMM 89*, September 1989.

[7] M. Shreedhar and G. Varghese, "Efficient Fair Queuing using Deficit Round Robin," *In Proc. of ACM SIGCOMM 95*, October 1995.

[8] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," *Proceedings of the IEEE*, pp. 1374-1396, October 1995.

[9] D. Bertsekas and R. Gallagher, *Data Networks,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall 1992.

[10] N. Abramson, "Development of the ALOHANET," *IEEE Transactions on Info. Theory*, vol. IT-25, no. 5, Sept. 1979.

[11] J. Limb and D. Sala, "A protocol for efficient transfer of Data over Hybrid Fiber Coax Systems," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 872-881, December 1997.

[12] Cable labs, *DOCSIS Radio Frequency Interface Specification*, v2.0, 2002.

[13] A. Tanenbaum, *Computer Networks*, 3rd ed., PrenticeHall, U saddle river, NJ, 1996.

[14] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, Vol. 3. No. 3, pp. 226-244, June 1995.

[15] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level" *In Proc. of the ACM/SIGCOMM*, 1995.

[16] W. Stallings, *Local & Metropolitan area networks*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[17] A. Gummalla and J. Limb, "Design of an access mechanism for a High Speed Distributed Wireless LAN," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, September 2000.

[18] IEEE 802.11 Working Group, *IEEE 802.11 standard for wireless LANs*, 1999.

[19] Y. Ma, P. So, Y. Guan, E. Gunawan, K. See, S. Chen and T. Lie, "Evaluation of MAC Protocols for Broadband PLC Networks", *In Proc. 6th International Power Engineering Conference (IPEC2003)*, Singapore, Nov. 2003.

[20] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Communications of ACM,* vol. 19, no. 7, July 1976.

[21] S. Kangude, M. Sherman, and J. Copeland, "An analysis of the Home PNA collision resolution mechanism," *Proc. IEEE LCN' 2003*, Bonn-Konigswinter, Oct. 2003.

[22] H. Peyravi, "Medium access control protocols performance in satellite communications," *IEEE Commun. Magazine*, vol. 37, no. 3, pp. 62-71, Mar. 1999.

[23] W. Yin and Y. Lin, "Statistically Optimized Minislot Allocation for Initial and Collision Resolution in Hybrid Fiber Coaxial Networks," *IEEE journal on selected areas in communications,* vol. 18, No. 9, September 2000.

[24] Z. Chang, I. Lee, C. Chang, C. Li, and B. Sui, "A novel scheme using the information of departure processes for delay guarantees of distributed VBR traffic," *IEEE Trans. on Networking*, vol. 9, no. 4, pp. 452-463, 2001.

[25] IEEE 802.11E Working Group, *IEEE 802.11e draft standard for QoS in wireless LANs*, Draft available from the working group.

[26] C. Foh and M. Zukerman, "CSMA with reservations by interruptions (CSMA/RI): A novel approach to reduce collisions in CSMA/CD," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, September 2000.

[27] N. Vaidya, P. Bahl, and Seema Gupta, "Distributed fair scheduling in a wireless LAN," *Proc. of the 6th annual International Conference on Mobile Computing and Networking*, p.167-178, August 2000.

[28] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1353-1368, August 1999.

[29] Y. Shu, Z. Jin, L. Zhang, L. Wang, and O. Yang, "Traffic Prediction using FARIMA models," *In Proc. of ICC'99*, 1999.

[30] F. Abi-Nassif and W. Lee, "Offered load estimation in a multimedia cable network system," in *Proc. ICC'99*, Vancouver, Canada, June 1999.

[31] N. Golmie, Y. Saintillan, and D. Su, "A review of contention resolution algorithms for IEEE 802.14 networks," *IEEE Communications Surveys,* first quarter 1999. Available: http://www.comsoc.org/pubs/surveys.

[32] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. IT-25, no. 5, September 1979.

[33] M. Sherman, "Method And System For Optimally Serving Stations On Wireless LANs Using A Controlled Contention/Resource Reservation Protocol Of The IEEE 802.11E Standard," US Patent application dated 8/29/2002.

[34] S. Lam, "Adaptive Backoff Algorithms for Multiple Access: A History," online article available at http://www.cs.utexas.edu/users/lam/NRL/backoff.html.

[35] W. Xu and G. Campbell, "A Distributed Queuing Random Access Protocol for a Broadcast Channel," *Proc. ACM SIGCOMM' 93*, San Francisco, Sept. 1993.

[36] S. Kangude, M. Sherman, and J. Copeland, "Optimality Analysis of the Home PNA Collision Resolution Mechanism," *Georgia Tech CSC Technical Report GIT-CSC-04-001*, January 2004.

[37] S. Kangude, M. Sherman, and J. Copeland, "Optimality Analysis of an Efficient N-ary Tree Contention Resolution Mechanism for CSMA Local Networks," *Georgia Tech CSC Technical Report GIT-CSC-04-002*, January 2004.

[38] M. Corner, J. Liebeherr, N. Golmie, C. Bisdikian, and D. Su. "A Priority Scheme for the IEEE 802.14 MAC Protocol for Hybrid Fiber-Coax Networks". *IEEE Transactions on Networking*, April, 2000.

[39] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, Vol. 5. No. 6, pp. 835-846, December 1997.

[40] W. E. Leland, M. S. Taqqu, W. Willinger, and Daniel V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended version)", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, February 1994.

[41] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” *Proc. SIGCOMM '94 Conference*, 1994.

[42] C. Wu and V. Li, “Receiver –initiated busy tone multiple access in packet radio networks,” *In Proc. SIGCOMM’88*, pp. 336-342.

[43] http://www.opnet.com

[44] J. Limb, “Performance evaluation process for MAC proto-cols,” *IEEE 802.14 Working Group Meet*., May 1996. contract IEEE802.14-96/083R2.

[45] R. Jain, D. Chiu, W. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,” *DEC Technical Report, DEC-TR-301*, September 1984.

[46] R. Addie, M. Zukerman, and T. Neame, “Broadband Traffic Modeling: Simple solutions to hard problems,” *IEEE Communications Magazine*, pp. 88-95, August 1998.

[47] D. Jagerman, B. Melamed, and W. Willinger, “Stochastic Modeling Of Traffic Processes,” *In Frontiers in Queuing: Models, Methods and Problems (J. Dshalalow, Edr.)*, CRC Press, 1996.

VITA

Shantanu Kangude was brought up in the city of Bhopal in India. He received his Bachelor of Technology degree in Electrical Engineering, with honors, from the Indian Institute of Technology in Kanpur, India. Thereafter, he joined the Georgia Institute of Technology (Georgia Tech) in Atlanta, USA, for his graduate studies. He received his Master of Science degree in Electrical and Computer Engineering from Georgia Tech in December 2000. He pursued his doctoral studies at the Communications Systems Center (CSC) at Georgia Tech. During the course of his doctoral research, he also interned at Intel labs and AT & T labs in Oregon, USA and New Jersey, USA respectively. During the summer of 2002, he interned at Lancope, Inc. in Georgia USA. In July 2004, he received his Doctor of Philosophy degree in Electrical and Computer Engineering from Georgia Tech. His research interests include computer networks quality of service, medium access protocols, local and metro area networks, network security, and wireless networks.